[Paper review 20]

Uncertainty in Deep Learning - Chapter 3

(Yarin Gal, 2016)

[Contents]

3. Bayesian Deep Learning

Based on two works

- 1) MC estimation (Graves, 2011)
- 2) VI (Hinton and Van Camp, 1993)

in a Bayesian persepective!

"BNN inference + SRTs (offer a practical inference technique)"

Steps

- step 1) analyze the variance of several stochastic estimators (used in VI)
- step 2) tie these derivations to SRTs
- step 3) propose practical techniques to obtain model uncertainty

3.1 Advanced techniques in VI

[review of VI]

$\mathcal{L}_{\mathrm{VI}}(\theta):=-\sum_{i=1}^{N} \int q_{\theta}(\boldsymbol{\omega}) \log p\left(\mathbf{y}_{i} \mid \mathbf{f}^{\omega}\left(\mathbf{x}_{i}\right)\right) \mathrm{d} \boldsymbol{\omega}+\mathrm{KL}\left(q_{\theta}(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right)$

expected log likelihood : $-\sum_{i=1}^{N} \int q_{\theta}(\boldsymbol{\omega}) \log p\left(\mathbf{y}_{i} \mid \mathbf{f}^{\omega}\left(\mathbf{x}_{i}\right)\right) \mathrm{d} \boldsymbol{\omega}$

problems in expected log likelihood :

- problem 1) $\sum_{i=1}^{N}$: perform computations over the entire dataset

- problem 2) $\int q_\theta(\boldsymbol{\omega}) \log p\left(\mathbf{y}_i \mid \mathbf{f}^\omega\left(\mathbf{x}_i\right)\right) \mathrm{d} \boldsymbol{\omega}$: not tractable

Solutions

- solution 1) data sub-sampling (mini-batch optimization)

  (unbiased + stochastic estimator)

  $\widehat{\mathcal{L}}_{\mathrm{VI}}(\theta):=-\frac{N}{M} \sum_{i \in S} \int q_\theta(\boldsymbol{\omega}) \log p\left(\mathbf{y}_i \mid \mathbf{f}^\omega\left(\mathbf{x}_i\right)\right) \mathrm{d} \boldsymbol{\omega}+\mathrm{KL}\left(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right)$
- solution 2) MC integration

3.1.1 MC estimators

use MC estimation to estimate "EXPECTED LOG LIKELIHOOD"

(more importantly, the "derivatives" of expected log likelihood)

Estimate "integral derivatives"

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

THREE main techniques for MC estimation for $\theta=\{\mu, \sigma\}$

- find "mean" derivative estimator
- find "standard deviation" derivative estimator

(1) Score function estimator

(= Likelihood ratio estimator, Reinforce)

(2) Path-wise derivative estimator

(= reparametrization trick)

(3) Characteristic function estimator

(1) Score function estimator (= $\hat{I}_1(\theta)$)

$\hat{I}_1(\theta)=f(x) \frac{\partial \log p_\theta(x)}{\partial \theta}$ with $x \sim p_\theta(x)$

- simple
- but high variance

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

(2) Path-wise derivative estimator (= $\hat{I}_2(\theta)$)

$\hat{I}_2(\theta)=f^{\prime}(g(\theta, \epsilon)) \frac{\partial}{\partial \theta} g(\theta, \epsilon)$

- $\hat{I}_2(\mu)=f^{\prime}(x)$
- $\hat{I}_2(\sigma)=f^{\prime}(x) \frac{(x-\mu)}{\sigma}$

reparameterization trick

- before) $p_\theta(x)=\mathcal{N}\left(x;\mu,\sigma^2\right)$
- after) $g(\theta,\epsilon)=\mu+\sigma\epsilon$ & $p(\epsilon)=\mathcal{N}(\epsilon;0,I)$

"mean" derivative estimator : $\frac{\partial}{\partial\mu}\int f(x)p_\theta(x)\mathrm{d}x=\int f'(x)p_\theta(x)\mathrm{d}x$

"standard deviation" derivative estimator : $\frac{\partial}{\partial\sigma}\int f(x)p_\theta(x)\mathrm{d}x=\int f'(x)\frac{(x-\mu)}{\sigma}p_\theta(x)\mathrm{d}x$

(3) Characteristic function estimator (= $\hat{I}_3(\theta)$)

$\hat{I}_2(\mu)=f'(x)$

$\hat{I}_3(\sigma)=\sigma f''(x)$ ($\frac{\partial}{\partial\sigma}\int f(x)p_\theta(x)\mathrm{d}x=2\sigma\cdot\frac{1}{2}\int f''(x)p_\theta(x)\mathrm{d}x$)

- rely on the characteristic function of "Gaussian distribution"

  (restricts the estimator to Gaussian $p_\theta(x)$ alone)

[tip] Reparameterization Trick

- use $p_\theta(x)=\int p_\theta(x,\epsilon)\mathrm{d}\epsilon=\int p_\theta(x\mid\epsilon)p(\epsilon)d\epsilon$
- where $p_\theta(x\mid\epsilon)=\delta(x-g(\theta,\epsilon))$

  (+ $\delta(x-g(\theta,\epsilon))$ is zero for all x apart from $x=g(\theta,\epsilon)$)

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\int f(x)p_\theta(x)\mathrm{d}x &= \frac{\partial}{\partial\theta}\int f(x)\left(\int p_\theta(x,\epsilon)\mathrm{d}\epsilon\right)\mathrm{d}x \\
&= \frac{\partial}{\partial\theta}\int f(x)p_\theta(x\mid\epsilon)p(\epsilon)\mathrm{d}\epsilon\mathrm{d}x \\
&= \frac{\partial}{\partial\theta}\int\left(\int f(x)\delta(x-g(\theta,\epsilon))\mathrm{d}x\right)p(\epsilon)\mathrm{d}\epsilon \\
&= \frac{\partial}{\partial\theta}\int f(g(\theta,\epsilon))p(\epsilon)\mathrm{d}\epsilon \\
&= \int\frac{\partial}{\partial\theta}f(g(\theta,\epsilon))p(\epsilon)\mathrm{d}\epsilon \\
&= \int f'(g(\theta,\epsilon))\frac{\partial}{\partial\theta}g(\theta,\epsilon)p(\epsilon)\mathrm{d}\epsilon
\end{aligned}
$$

3.1.2 Variance Analysis of MC estimator

None of 3 estimators has the lowest variance for all functions of $f(x)$

- (1) score function
- (2) path-wise derivative function
- (3) characteristic function

Properties that (2), (3) have lower variance than (1) : in the paper

From empirical observations, (2) seems to be good!

Will continue our work using the path-wise derivative estimator

3.2 Practical Inference in BNN

in terms of "PRACTICALITY"

Graves (2011)

- (a) delta approximating distribution (use "characteristic function")
- (b) fully factorized approximating distribution

  (factorized the approximating distribution for EACH WEIGHT scalar, thus "losing weight correlation" $\rightarrow$ hurt performance)

Advancement

- (a) use "path-wise derivative estimator" instead (used 're-parameterization trick ')
- (b) factorize the approximating distribution for EACH ROW WEIGHT

ELBO using (1) reparam trick & (2) MC estimation

$$
\begin{aligned}
\widehat{\mathcal{L}}_{\mathrm{VI}}(\theta) &= -\frac{N}{M} \sum_{i \in S} \int q_\theta(\boldsymbol{\omega}) \log p\left(\mathbf{y}_i \mid \mathbf{f}^\omega\left(\mathbf{x}_i\right)\right) \mathrm{d}\boldsymbol{\omega} + \mathrm{KL}\left(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right) \\
&= -\frac{N}{M} \sum_{i \in S} \int p(\boldsymbol{\epsilon}) \log p\left(\mathbf{y}_i \mid \mathbf{f}^{g(\theta, \boldsymbol{\epsilon})}\left(\mathbf{x}_i\right)\right) \mathrm{d}\boldsymbol{\epsilon} + \mathrm{KL}(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})) \\
&\approx -\frac{N}{M} \sum_{i \in S} \log p\left(\mathbf{y}_i \mid \mathbf{f}^{g(\theta, \epsilon)}\left(\mathbf{x}_i\right)\right) + \mathrm{KL}\left(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right) \\
&\equiv \widehat{\mathcal{L}}_{\mathrm{MC}}(\theta) \qquad \text{where } \mathbb{E}_{S, \epsilon}\left(\widehat{\mathcal{L}}_{\mathrm{MC}}(\theta)\right) = \mathcal{L}_{\mathrm{VI}}(\theta)
\end{aligned}
$$

Predictive distribution

$$
\begin{aligned}
\tilde{q}_\theta\left(\mathbf{y}^* \mid \mathbf{x}^*\right) := \frac{1}{T} \sum_{t=1}^T p\left(\mathbf{y}^* \mid \mathbf{x}^*, \widehat{\boldsymbol{\omega}}_t\right) \underset{T \rightarrow \infty}{\longrightarrow} & \int p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}\right) q_\theta(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
\approx & \int p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}\right) p(\boldsymbol{\omega} \mid \mathbf{X}, \mathbf{Y}) \mathrm{d}\boldsymbol{\omega} \\
= & \, p\left(\mathbf{y}^* \mid \mathbf{x}^*, \mathbf{X}, \mathbf{Y}\right)
\end{aligned}
$$

where $\widehat{\omega}_t \sim q_\theta(\omega)$

[Summary]

optimizing $\widehat{\mathcal{L}}_{\mathrm{MC}}(\theta)$ w.r.t θ = optimizing $\widehat{\mathcal{L}}_{\mathrm{VI}}(\theta)$ w.r.t θ

Algorithm 1 Minimise divergence between $q_\theta(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega}|X, Y)$

1: Given dataset $\mathbf{X}, \mathbf{Y}$,
2: Define learning rate schedule η,
3: Initialise parameters θ randomly.
4: **repeat**
5: Sample M random variables $\widehat{\boldsymbol{\epsilon}}_i \sim p(\boldsymbol{\epsilon})$, S a random subset of $\{1, .., N\}$ of size M.
6: Calculate stochastic derivative estimator w.r.t. θ:

$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M} \sum_{i \in S} \frac{\partial}{\partial\theta} \log p(\mathbf{y}_i | \mathbf{f}^{g(\theta, \widehat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)) + \frac{\partial}{\partial\theta} \mathrm{KL}(q_\theta(\boldsymbol{\omega}) || p(\boldsymbol{\omega})).$$

7: Update θ:

$$\theta \leftarrow \theta + \eta \widehat{\Delta\theta}.$$

8: **until** θ has converged.

3.2.1 SRT (Stochastic Regularization Techniques)

"REGULARIZE models through injection of STOCHASTIC NOISE"

ex) dropout, multiplicative Gaussian Noise, drop connect ...

notation

- M : deterministic matrix
- W : random variable defined over the set of real matrices
- $\hat{W}$: realization of W

(1) Dropout

- two binary vectors $\hat{\epsilon}_1, \hat{\epsilon}_2$

  (each with dimension Q (=input dim) and K (=intermediate dim))
- parameters : $\theta = \{M_1, M_2, b\}$
- $\hat{\mathbf{y}} = \widehat{\mathbf{h}}\mathbf{M}_2$
  - $\widehat{\mathbf{h}} = \mathbf{h} \odot \hat{\epsilon}_2$
    - $\mathbf{h} = \sigma(\widehat{\mathbf{x}}\mathbf{M}_1 + \mathbf{b})$
      - $\widehat{\mathrm{x}} = \mathrm{x} \odot \hat{\epsilon}_1^3$
- sample $\hat{\epsilon_i}$ for every input & every forward pass

  use the same value for backward pass
- test time : do not sample any variables & just use the original units $\mathbf{x}, \mathbf{h}$ scaled by $\frac{1}{1-p_i}$

(2) Multiplicative Gaussian Noise

- same as (1), except $\hat{\epsilon_i} \sim N(1, \alpha)$

3.2.2 SRT as approximate inference

inject noise to feature space (=the input features to each layer, which are x and h)

$$\begin{aligned}
\hat{\mathbf{y}} &= \widehat{\mathbf{h}}\mathbf{M}_2 \\
&= (\mathbf{h} \odot \hat{\boldsymbol{\epsilon}}_2)\,\mathbf{M}_2 \\
&= (\mathbf{h} \cdot \operatorname{diag}(\hat{\boldsymbol{\epsilon}}_2))\,\mathbf{M}_2 \\
&= \mathbf{h}\,(\operatorname{diag}(\hat{\boldsymbol{\epsilon}}_2)\mathbf{M}_2) \\
&= \sigma(\hat{\mathbf{x}}\mathbf{M}_1 + \mathbf{b})\,(\operatorname{diag}(\hat{\boldsymbol{\epsilon}}_2)\mathbf{M}_2) \\
&= \sigma((\mathbf{x} \odot \hat{\boldsymbol{\epsilon}}_1)\,\mathbf{M}_1 + \mathbf{b})\,(\operatorname{diag}(\hat{\boldsymbol{\epsilon}}_2)\mathbf{M}_2) \\
&= \sigma(\mathbf{x}\,(\operatorname{diag}(\hat{\boldsymbol{\epsilon}}_1)\mathbf{M}_1) + \mathbf{b})\,(\operatorname{diag}(\hat{\boldsymbol{\epsilon}}_2)\mathbf{M}_2) \\
&= \sigma\left(\mathbf{x}\widehat{\mathbf{W}}_1 + \mathbf{b}\right)\widehat{\mathbf{W}}_2 =: \mathbf{f}^{\widehat{\mathbf{W}}_1}, \widehat{\mathbf{W}}_2, \mathbf{b}(\mathbf{x})
\end{aligned}$$

(let $\widehat{\mathbf{W}}_1 := \operatorname{diag}(\hat{\epsilon}_1)\mathbf{M}_1$ and $\widehat{\mathbf{W}}_2 := \operatorname{diag}(\hat{\epsilon}_2)\mathbf{M}_2$)

(random variable realization as weights : $\widehat{\omega} = \left\{\widehat{\mathbf{W}}_1, \widehat{\mathbf{W}}_2, \mathbf{b}\right\}$)

Loss function

$\widehat{\mathcal{L}}_{\text{dropout}}\ (\mathbf{M}_1, \mathbf{M}_2, \mathbf{b}) := \frac{1}{M}\sum_{i \in S} E^{\widehat{\mathbf{W}}_1^i, \widehat{\mathbf{W}}_2^i, \mathbf{b}}\ (\mathbf{x}_i, \mathbf{y}_i) + \lambda_1 \|\mathbf{M}_1\|^2 + \lambda_2 \|\mathbf{M}_2\|^2 + \lambda_3 \|\mathbf{b}\|^2$

where $\widehat{\mathbf{W}}_1^i, \widehat{\mathbf{W}}_2^i$ corresponding to new masks $\hat{\epsilon}_1^i, \hat{\epsilon}_2^i$

Example) Negative Log Likelihood

$E^{\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}}(\mathrm{x},\mathrm{y})=\frac{1}{2}\left\|\mathrm{y}-\mathrm{f}^{\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}}(\mathrm{x})\right\|^2=-\frac{1}{\tau}\log p\left(\mathrm{y}\mid \mathrm{f}^{\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}}(\mathrm{x})\right)+\mathrm{const}$

- where $p\left(\mathbf{y}\mid \mathbf{f}^{\mathbf{M}_1,\mathbf{M}_2,\mathbf{b}}(\mathbf{x})\right)=\mathcal{N}\left(\mathbf{y};\mathbf{f}^{\mathbf{M}_1,\mathbf{M}_2,\mathbf{b}}(\mathbf{x}),\tau^{-1}I\right)$ with τ^{-1} observation noise
- (for classification) $\tau=1$

Reparametrization trick

$\widehat{\omega}_i=\left\{\widehat{\mathbf{W}}_1^i,\widehat{\mathbf{W}}_2^i,\mathbf{b}\right\}=\left\{\operatorname{diag}\left(\hat{\epsilon}_1^i\right)\mathbf{M}_1,\operatorname{diag}\left(\hat{\epsilon}_2^i\right)\mathbf{M}_2,\mathbf{b}\right\}=: g\left(\theta,\hat{\epsilon}_i\right)$

where $\theta=\{\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}\}$, $\hat{\epsilon}_1^i\sim p\left(\epsilon_1\right)$, and $\hat{\epsilon}_2^i\sim p\left(\epsilon_2\right)$ for $1\leq i\leq N$

Loss function :

$$\widehat{\mathcal{L}}_{\text{dropout}}\left(\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}\right)=-\frac{1}{M\tau}\sum_{i\in S}\log p\left(\mathbf{y}_i\mid \mathbf{f}^{g\left(\theta,\hat{\epsilon}_i\right)}(\mathrm{x})\right)+\lambda_1\left\|\mathrm{M}_1\right\|^2+\lambda_2\left\|\mathrm{M}_2\right\|^2+\lambda_3\|\ \mathrm{b}\|^2$$

Derivative of the loss function :

$$\frac{\partial}{\partial\theta}\widehat{\mathcal{L}}_{\text{dropout}}(\theta)=-\frac{1}{M\tau}\sum_{i\in S}\frac{\partial}{\partial\theta}\log p\left(\mathbf{y}_i\mid \mathbf{f}^{g\left(\theta,\hat{\epsilon}_i\right)}(\mathbf{x})\right)+\frac{\partial}{\partial\theta}\left(\lambda_1\left\|\mathbf{M}_1\right\|^2+\lambda_2\left\|\mathbf{M}_2\right\|^2+\lambda_3\|\mathbf{b}\|^2\right)$$

[Summary]

optimizing $\widehat{\mathcal{L}}_{\text{dropout}}\left(\mathrm{M}_1,\mathrm{M}_2,\ \mathrm{b}\right)$ with dropout :

Algorithm 2 Optimisation of a neural network with dropout

1: Given dataset $\mathbf{X},\mathbf{Y}$,
2: Define learning rate schedule η,
3: Initialise parameters θ randomly.
4: **repeat**
5: Sample M random variables $\widehat{\boldsymbol{\epsilon}}_i\sim p(\boldsymbol{\epsilon})$, S a random subset of $\{1,..,N\}$ of size M.
6: Calculate derivative w.r.t. θ:

$$\widehat{\Delta\theta}\leftarrow-\frac{1}{M\tau}\sum_{i\in S}\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{f}^{g(\theta,\widehat{\epsilon}_i)}(\mathbf{x}))+\frac{\partial}{\partial\theta}\Big(\lambda_1||\mathbf{M}_1||^2+\lambda_2||\mathbf{M}_2||^2+\lambda_3||\mathbf{b}||^2\Big).$$

7: Update θ:

$$\theta\leftarrow\theta+\eta\widehat{\Delta\theta}.$$

8: **until** θ has converged.

instead of Dropout...

1) Multiplicative Gaussian Noise

- $g(\theta,\boldsymbol{\epsilon})=\left\{\operatorname{diag}\left(\boldsymbol{\epsilon}_1\right)\mathbf{M}_1,\operatorname{diag}\left(\boldsymbol{\epsilon}_2\right)\mathbf{M}_2,\mathbf{b}\right\}$

  with $p\left(\boldsymbol{\epsilon}_l\right)$(for $l=1,2$) a product of $\mathcal{N}(1,\alpha)$ with positive α

2) Drop connect

- $g(\theta,\boldsymbol{\epsilon})=\left\{\mathbf{M}_1\odot\boldsymbol{\epsilon}_1,\mathbf{M}_2\odot\boldsymbol{\epsilon}_2,\mathbf{b}\right\}$

  with $p\left(\epsilon_l\right)$ a product of Bernoulli random variables

3) Additive Gaussian Noise

- $g(\theta, \boldsymbol{\epsilon})=\{\mathbf{M}_1+\boldsymbol{\epsilon}_1, \mathbf{M}_2+\boldsymbol{\epsilon}_2, \mathbf{b}\}$
  with $p\left(\boldsymbol{\epsilon}_l\right)$ a product of $\mathcal{N}(0, \alpha)$ for each weight scalar

[Algorithm summary]

[3.2.1] Algorithm 1) Minimize divergence between $q_\theta(w)=p(w \mid X, Y)$

[3.2.2] Algorithm 2) Optimization of a NN with Dropout

For algorithm 1 = 2....

- 1) "regularization term derivatives" should be same (= KL condition)
  $\frac{\partial}{\partial \theta} \mathrm{KL}\left(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right)=\frac{\partial}{\partial \theta} N \tau\left(\lambda_1\left\|\mathbf{M}_1\right\|^2+\lambda_2\left\|\mathbf{M}_2\right\|^2+\lambda_3\|\mathbf{b}\|^2\right)$
- 2) scale of derivatives
  $\frac{\partial}{\partial \theta} \widehat{\mathcal{L}}_{\text {dropout }}(\theta)=\frac{1}{N \tau} \frac{\partial}{\partial \theta} \widehat{\mathcal{L}}_{\mathrm{MC}}(\theta)$

Summary :

- "Optimizing any NN with DROPOUT" = "APPROXIMATE INFERENCE in a probabilistic interpretation of the model"
- NN trained with dropout is "Bayesian NN"

3.2.3 KL condition

condition for "VI" = "DO" $\rightarrow$ depends on the model specification (choice of $p(w)$ and $q_\theta(w)$)

Example 1)

- prior : $p(\boldsymbol{\omega})=\prod_{i=1}^L p\left(\mathbf{W}_i\right)=\prod_{i=1}^L \mathcal{N}\left(0, \mathbf{I} / l_i^2\right)$, where $l_i^2=\frac{2 N \tau \lambda_i}{1-p_i}$ (prior length scale)
- then
  $\frac{\partial}{\partial \theta} \mathrm{KL}\left(q_\theta(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})\right) \approx \frac{\partial}{\partial \theta} N \tau\left(\lambda_1\left\|\mathbf{M}_1\right\|^2+\lambda_2\left\|\mathbf{M}_2\right\|^2+\lambda_3\|\mathbf{b}\|^2\right)$

Example 2) discrete prior

- $p(\mathbf{w}) \propto e^{-\frac{l^2}{2} \mathbf{w}^T \mathbf{w}}$

Example 3) improper log-uniform prior

- for Multiplicative Gaussian Noise

3.3 Model Uncertainty in BNN

approximate predictive distribution

$q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right)=\int p\left(\mathbf{y}^* \mid \mathbf{f}^\omega\left(\mathbf{x}^*\right)\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d} \boldsymbol{\omega}$

- $\omega=\left\{\mathrm{W}_i\right\}_{i=1}^L$ is our set of random variables for a model with L layers
- f^ω : model's stochastic output
- $q_\theta^*(\omega)$: optimum

check if FIRST & SECOND MOMENT matches!

First Moment

$\widetilde{\mathbb{E}}\left[\mathbf{y}^*\right] := \frac{1}{T}\sum_{t=1}^{T} \mathrm{f}^{\widehat{\omega}t}\left(\mathbf{x}^*\right) \underset{T\to\infty}{\to} \mathbb{E}q_{q_\theta^*\left(\mathrm{y}^* \mid \mathrm{x}^*\right)}\left[\mathrm{y}^*\right]$

with $\widehat{\boldsymbol{\omega}}_t \sim q_\theta^*(\boldsymbol{\omega})$

- unbiased estimator, following MC integration with T samples
- when use Dropout $\rightarrow$ "MC Dropout" (= model averaging)

(proof)

$$\begin{aligned}
\mathbb{E}_{q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right)}\left[\mathbf{y}^*\right] &= \int \mathbf{y}^* q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right) \mathrm{d}\mathbf{y}^* \\
&= \iint \mathbf{y}^* \mathcal{N}\left(\mathbf{y}^*; \mathbf{f}^{\boldsymbol{\omega}}\left(\mathbf{x}^*\right), \tau^{-1}\mathbf{I}\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \mathrm{d}\mathbf{y}^* \\
&= \int \left(\int \mathbf{y}^* \mathcal{N}\left(\mathbf{y}^*; \mathbf{f}^{\boldsymbol{\omega}}\left(\mathbf{x}^*\right), \tau^{-1}\mathbf{I}\right) \mathrm{d}\mathbf{y}^*\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int \mathbf{f}^{\omega}\left(\mathbf{x}^*\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}
\end{aligned}$$

Second Moment

$\widetilde{\mathbb{E}}\left[\left(\mathbf{y}^*\right)^T\left(\mathbf{y}^*\right)\right] := \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \mathbf{f}^{\widehat{\omega}_t}\left(\mathbf{x}^*\right)^T \mathbf{f}^{\widehat{\omega}_t}\left(\mathbf{x}^*\right) \underset{T\to\infty q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right)}{\mathbb{E}}\left[\left(\mathbf{y}^*\right)^T\left(\mathbf{y}^*\right)\right]$

with $\widehat{\boldsymbol{\omega}}_t \sim q_\theta^*(\boldsymbol{\omega})$ and $\mathrm{y}^*, \mathrm{f}\hat{\omega}_t\left(\mathrm{x}^*\right)$ row vectors

- unbiased estimator, following MC integration with T samples

(proof)

$$\begin{aligned}
\mathbb{E}_{q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right)}\left[\left(\mathbf{y}^*\right)^T\left(\mathbf{y}^*\right)\right] &= \int\left(\int\left(\mathbf{y}^*\right)^T\left(\mathbf{y}^*\right) p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}\right) \mathrm{d}\mathbf{y}^*\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int\left(\operatorname{Cov}_{p\left(\mathbf{y}^* \mid \mathbf{x}^*, \omega\right)}\left[\mathbf{y}^*\right] + \mathbb{E}_{p\left(\mathbf{y}^* \mid \mathbf{x}^*, \omega\right)}\left[\mathbf{y}^*\right]^T \mathbb{E}_{p\left(\mathbf{y}^* \mid \mathbf{x}^*, \omega\right)}\left[\mathbf{y}^*\right]\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int\left(\tau^{-1}\mathbf{I} + \mathbf{f}^{\omega}\left(\mathbf{x}^*\right)^T \mathbf{f}^{\boldsymbol{\omega}}\left(\mathbf{x}^*\right)\right) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}
\end{aligned}$$

Variance

$\widehat{\operatorname{Var}}\left[\mathbf{y}^*\right] := \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \mathbf{f}^{\widehat{\omega}_t}\left(\mathbf{x}^*\right)^T \mathbf{f}^{\widehat{\omega}_t}\left(\mathbf{x}^*\right) - \widetilde{\mathbb{E}}\left[\mathbf{y}^*\right]^T \widetilde{\mathbb{E}}\left[\mathbf{y}^*\right] \underset{T\to\infty}{\longrightarrow} \operatorname{Var}_{q_\theta^*\left(\mathbf{y}^* \mid \mathbf{x}^*\right)}\left[\mathbf{y}^*\right]$

How to find model precision τ ?

- with grid search, find weight-decay λ
- $\tau = \frac{(1-p)l_i^2}{2N\lambda_i}$

Predictive Log-likelihood

(approximated by MC integration)

$\log p\left(\mathbf{y}^* \mid \mathbf{x}^*, \mathbf{X}, \mathbf{Y}\right) := \widehat{\log\left(\frac{1}{T}\sum_{t=1}^{T} p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}_t\right)\right)} \underset{T\to\infty}{\longrightarrow}$

$\log \int p\left(\mathbf{y}^{*} \mid \mathbf{x}^{*}, \boldsymbol{\omega}\right) q_{\theta}^{*}(\boldsymbol{\omega}) \mathrm{d} \boldsymbol{\omega}$
$\approx \log \int p\left(\mathbf{y}^{*} \mid \mathrm{x}^{*}, \boldsymbol{\omega}\right) p(\boldsymbol{\omega} \mid \mathbf{X}, \mathbf{Y}) \mathrm{d} \boldsymbol{\omega}$
$=\log p\left(\mathbf{y}^{*} \mid \mathbf{x}^{*}, \mathbf{X}, \mathbf{Y}\right)$

- for regression : $\widetilde{\log} p\left(\mathbf{y}^{*} \mid \mathbf{x}^{*}, \mathbf{X}, \mathbf{Y}\right)=\operatorname{logsumexp}\left(-\frac{1}{2} \tau\left\|\mathbf{y}-\mathbf{f}^{\widehat{\omega}_{t}}\left(\mathbf{x}^{*}\right)\right\|^{2}\right)-\log T-\frac{1}{2} \log 2 \pi+\frac{1}{2} \log \tau$

3.3.1 Uncertainty in Classification

(regression) find predictive uncertainty by "looking at the sample variance of multiple stochastic forward pass"

(classification)

- 1) variation ratios
- 2) predictive entropy
- 3) mutual information

1) Variation Ratio

$\text{variation-ratio }[\mathrm{x}]:=1-\frac{f_{\mathrm{x}}}{T}$

- sample a label from softmax probabilities
- collecting a set of T labels y_t from multiple stochastic forward passes (of the same input)
- $f_{\mathrm{x}}=\sum_{t} 1\left[y^{t}=c^{*}\right]$, where $c^{*}=\underset{c=1, \ldots, C}{\arg \max } \sum_{t} 1\left[y^{t}=c\right]$

variation-ratio can be seen as approximating the quantity : $1-p\left(y=c^{*} \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right)$

2) Predictive entropy

$\mathbb{H}\left[y \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right]:=-\sum_{c} p\left(y=c \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right) \log p\left(y=c \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right)$

- foundation in "information theory"

  (captures the information contained in the predictive distribution)
- summing over all possible classes c that y can take

3) Mutual Information

$$\begin{aligned}
\mathbb{I}\left[y, \boldsymbol{\omega} \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right]:= & \mathbb{H}\left[y \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right]-\mathbb{E}_{p\left(\boldsymbol{\omega} \mid \mathcal{D}_{\text {train }}\right)}[\mathbb{H}[y \mid \mathbf{x}, \boldsymbol{\omega}]] \\
= & -\sum_{c} p\left(y=c \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right) \log p\left(y=c \mid \mathbf{x}, \mathcal{D}_{\text {train }}\right)+\mathbb{E}_{p\left(\boldsymbol{\omega} \mid \mathcal{D}_{\text {train }}\right)}\left[\sum_{c} p(y=c \mid \mathbf{x}, \boldsymbol{\omega}) \log p(y=c \mid \mathbf{x}, \boldsymbol{\omega})\right]
\end{aligned}$$

- mutual information between prediction y and posterior (over the w)

Example (with binary output)

- case 1) all equal to 1 ...... ((1,0), (1,0), ... (1,0))
- case 2) all equal to 0.5 ...... ((0.5,0.5), (0.5,0.5), ... (0.5,0.5))
- case 3) half 1, half 0 ...... ((1,0), (0,1), ... (1,0))

example	Predictive Uncertainty	Model Uncertainty
case 1	LOW (=0)	LOW (=0)
case 2	HIGH (=0.5)	LOW (=0)
case 3	HIGH (=0.5)	HIGH (=0.5)

in case 2)

- variation ratio & predictive entropy = 0.5
- mutual information = 0

3.3.2 Difficulties with the approach

simple! just several stochastic forward pass & find sample mean and variance

but have 3 shortcomings...

- 1) test time is scaled by T

  (but not a real concern in a real world application ... transferring an input to a GPU)
- 2) model's uncertainty is not calibrated

  (calibrated model : predictive probabilities match the empirical frequency of the data)

  (GP's uncertainty is known to not be calibrated)

  (lack of calibration = scale is different! can not compare...)
- 3) limitation of VI : underestimation of predictive variance

  (but not a real concern in a real world application)

3.4 Approximate inference in complex models

apply it to CNN & RNN

3.4.1 Bayesian CNN

- also apply dropout after all convolution layers

3.4.2 Bayesian RNN

- inference with Bernoulli variational distributions for RNNs

$\mathrm{f}_{\mathrm{y}}\left(\mathrm{h}_T\right)=\mathrm{h}_T \mathrm{~W}_{\mathrm{y}}+\mathrm{b}_{\mathrm{y}}$

where $\mathbf{h}_t=\mathbf{f}_{\mathbf{h}}\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right)=\sigma\left(\mathbf{x}_t \mathbf{W}_{\mathbf{h}}+\mathbf{h}_{t-1} \mathbf{U}_{\mathbf{h}}+\mathbf{b}_{\mathbf{h}}\right)$

- view the under RNN model as a probabilistic model

regard $\omega=\left\{\mathbf{W}_{\mathbf{h}}, \mathbf{U}_{\mathbf{h}}, \mathbf{b}_{\mathbf{h}}, \mathbf{W}_{\mathbf{y}}, \mathbf{b}_{\mathbf{y}}\right\}$

$$\begin{aligned}
\int q(\boldsymbol{\omega}) \log p\left(\mathbf{y} \mid \mathbf{f}_{\mathbf{y}}^\omega\left(\mathbf{h}_T\right)\right) \mathrm{d} \boldsymbol{\omega} & =\int q(\boldsymbol{\omega}) \log p\left(\mathbf{y} \mid \mathbf{f}_{\mathbf{y}}^\omega\left(\mathbf{f}_{\mathbf{h}}^\omega\left(\mathbf{x}_T, \mathbf{h}_{T-1}\right)\right)\right) \mathrm{d} \boldsymbol{\omega} \\
& =\int q(\boldsymbol{\omega}) \log p\left(\mathbf{y} \mid \mathbf{f}_{\mathbf{y}}^\omega\left(\mathbf{f}_{\mathbf{h}}^\omega\left(\mathbf{x}_T, \mathbf{f}_{\mathbf{h}}^\omega\left(\ldots \mathbf{f}_{\mathbf{h}}^\omega\left(\mathbf{x}_1, \mathbf{h}_0\right) \ldots\right)\right)\right)\right) \mathrm{d} \boldsymbol{\omega} \\
& \approx \log p\left(\mathbf{y} \mid \mathbf{f}_{\mathbf{y}}^{\widehat{\omega}}\left(\mathbf{f}_{\mathbf{h}}^{\widehat{\omega}}\left(\mathbf{x}_T, \mathbf{f}_{\mathbf{h}}^{\widehat{\omega}}\left(\ldots \mathbf{f}_{\mathbf{h}}^{\widehat{\omega}}\left(\mathbf{x}_1, \mathbf{h}_0\right) \ldots\right)\right)\right)\right)
\end{aligned}$$

where $\widehat{\omega} \sim q(\omega)$

Final objective function :

$$\widehat{\mathcal{L}}_{\mathrm{MC}}=-\sum_{i=1}^{N} \log p\left(\mathbf{y}_i \mid \mathbf{f}_{\mathbf{y}}^{\widehat{\omega}_i}\left(\mathbf{f}_{\mathbf{h}}^{\widehat{\omega}_i}\left(\mathbf{x}_{i,T}, \mathbf{f}_{\mathbf{h}}^{\widehat{\omega}_i}\left(\ldots \mathbf{f}_{\mathbf{h}}^{\widehat{\omega}_i}\left(\mathbf{x}_{i,1}, \mathbf{h}_0\right) \ldots\right)\right)\right)\right)+\mathrm{KL}(q(\boldsymbol{\omega}) \| p(\boldsymbol{\omega}))$$