[Paper review 42]

A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference

(Shridhar, et al., 2019)

[Contents]

1. Abstract

propose **Bayesian CNN using VI**

- introduce probability distribution over the weights
- model of BBB (Bayes by Backprop)

BBB

- variational approximation to true posterior
- two params : mean & var

Bayesian CNN

- achieves performances equivalent to frequentist inference
- obtain measurement for uncertainties & regularization

Finally, propose ways to **prune** the Bayesian architecture

(make more computational & time effective)

2. Introduction

- DNNs
- CNNs
- Various Regularization techniques (early stopping, weight deacy, L1, L2..)

2-1. Problem statement

DNN : over-confident decision

$\rightarrow$ introduce Bayesian learning to CNN, thus giving uncertainty estimation & regularization

2-2. Current situation

DNN is widely used in many domains (usually single point-estimates architecture)

Bayesian posterior Inference over NN, attractive for solving overfitting

BUT, **CNNs has naver been successful**... due to practical issues

- computationally expensive
- double the number of model params

2-3. Our Hypothesis

use BBB!

exact Bayesian inference : number of parameters is very large...

so, approximate with variational distn $q_\theta(\mathbf{w} \mid D)$

2-4. Our Contribution

- 1) BBB can be efficiently applied to CNNs
- 2) Richer representations and predictions from cheap model averaging
- 3) VI can be applied to various CNN arthictectures
- 4) Examine how to estimate "aleatoric" and "epistemic" uncertainties

- 5) Only doubles the number of params, but infinite ensemble! (using unbiased MC estimates of grads)
- 6) L1 norm for pruning

Summary : **BBB is now applicable to FC, RNN, CNN !**

3. Background

3-1. Neural Network

- skip

3-2. Probabilistic ML

VI

- skip

LRT

- Local reparameterization trick

  Type of reparameterization, when the global uncertainty in the weights is translated into a form of "local uncertainty" which is independent across examples

3-3. Uncertainties in Bayesian Learning

2 types of uncertainties

- 1) Aleatoric : noise inherent in data... can not be reduced by further data collection
  - 1-1) Homoscedastic (uncertainty stays constant for different input)
  - 1-2) Heteroscedastic (uncertainty differs for different input)
- 2) Epistemic : casused by model ... can be reduced, given more data

Lots of works measures uncertainties by placing probabilities over....

- 1) model parameter (when dealing with "Epistemic uncertainty")
- 2) model output (when dealing with "Aleatoric uncertainty")

3-4. BBB (Bayes by Backprop)

VI method to learn posterior on the weights $w \sim q_\theta(w \mid \mathcal{D})$

Regularize the weights, by minimizing (negative) ELBO (or Variational Free energy)

optimal parameters :

$$\begin{aligned}\theta^{opt} &= \arg\min_{\theta} \mathrm{KL}\left[q_\theta(w \mid \mathcal{D}) \| p(w \mid \mathcal{D})\right] \\ &= \arg\min_{\theta} \mathrm{KL}\left[q_\theta(w \mid \mathcal{D}) \| p(w)\right] - \mathbb{E}_{q(w \mid \theta)}[\log p(\mathcal{D} \mid w)] + \log p(\mathcal{D})\end{aligned}.$$

Variational Free energy

- negative ELBO
- 1st part) $\mathrm{KL}\left[q_\theta(w \mid \mathcal{D}) \| p(w)\right]$: dependent on prior ..... called "complexity cost"

  2nd part) $\mathbb{E}_{q(w \mid \theta)}[\log p(\mathcal{D} \mid w)]$: dependent on data $p(D \mid w)$ ..... called "likelihood cost"

Use stochastic variational method

$\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^{n} \log q_\theta\left(w^{(i)} \mid \mathcal{D}\right) - \log p\left(w^{(i)}\right) - \log p\left(\mathcal{D} \mid w^{(i)}\right)$.

- sample $w^{(i)}$ from $q_\theta(w \mid D)$

3-5. Model weights pruning

Model pruning

- reduces sparsity in VNN
- thus, **reduce the number of valued parameters**

  (without much loss in the accuracy)

Several ways to prune model

- 1) most popular : low contributing weights $\rightarrow$ make 0

  (= L_0 norm , where $L_0 = \|\theta\|_0 = \sum_j \delta\left(\theta_j \neq 0\right)$ ... constant penalty to all non-zero weights)
- 2) alternative : L_1 norm, which this paper uses

  ($\|\theta\|_1 = \sum_j \left|\theta_j\right|. L_1$)

4. Related work

4-1. Bayesian Training

Applying Bayesian methods to NN

how to deal with intractable posterior $p(w \mid D)$>

- 1) MAP schemes for NN
- 2) Variational methods, natural regularizer (Hinton and Van Camp, 1993)
- 3) Laplace approximation
- 4) HMC for training NN
- 5) VI for NN
- 6) Dropout & Gaussian Dropout

4-2. Uncertainty Estimation

has not been successful until 2015

Dropout as a Bayesian approximation: Insights and applications (Gal and Ghahramani, 2015)

- NN trained with dropout = approximate Bayesian model
- uncertainty can be obtained by computing the **variance on multiple predictions** with different **dropout masks**

5. Our concept

5-1. Bayesian CNN with VI

Local Reparameterization Trick for Convolutional Layers

apply LRT to CNNs

(do not sample from weights w, but sample from layer activations b)

variational posterior distn : $q_\theta\left(w_{ijhw} \mid \mathcal{D}\right)=\mathcal{N}\left(\mu_{ijhw}, \alpha_{ijhw} \mu_{ijhw}^2\right)$

LRT :$b_j=A_i * \mu_i+\epsilon_j \odot \sqrt{A_i^2 *\left(\alpha_i \odot \mu_i^2\right)}$. where $\epsilon_j \sim \mathcal{N}(0,1), A_i$

- A_i : receptive field
- $*$: convolutional operation
- $\odot$: component-wise multiplication

Applying two Sequential Convolutional Operations (for mean & var)

(original CNN) single point estimate : **one** convolutional operations

(Bayesian CNN) single point estimate : **two** convolutional operations

output b is a function of...

- mean : μ_{ijwh}
- variance : $\alpha_{ijhw}\mu_{ijhw}^2$

Two convolutional operations

- step 1) treat b as an output of CNN as frequentists & just optimize

  interpret this single point estimate as mean (μ_{ijwh})
- step 2) learn variance $\alpha_{ijhw}\mu_{ijhw}^2$

  (have learned mean in step1, so only need to learn α_{ijhw})

Summary

- in the first convolutional operation, learn MAP of variational posterior distn $q_\theta(w \mid D)$
- in the second convolutional operation, observe how much values for weights w deviate from this MAP
- TIP
  - to ensure non-zero variance & to enhance accuarcy...

    learn $\log \alpha_{ijhw}$ and use Softplus activation function

5-2. Uncertainty Estimation in CNN

Predictive distribution :

$p_{\mathcal{D}}\left(y^* \mid x^*\right)=\int p_w\left(y^* \mid x^*\right) p_{\mathcal{D}}(w) d w .$

BBB

- $q_\theta(w \mid \mathcal{D}) \sim \mathcal{N}\left(w \mid \mu, \sigma^2\right)$

  where $\theta=\{\mu, \sigma\}$ are learned
- for classification...

$$\begin{aligned} p_{\mathcal{D}}\left(y^* \mid x^*\right) & =\int \operatorname{Cat}\left(y^* \mid f_w\left(x^*\right)\right) \mathcal{N}\left(w \mid \mu, \sigma^2\right) d w \\ & =\int \prod_{c=1}^C f\left(x_c^* \mid w\right)^{y_c^*} \frac{1}{\sqrt{2 \pi \sigma^2}} e^{-\frac{(w-\mu)^2}{2 \sigma^2}} d w \end{aligned} .$$

No closed-form (no conjugacy between categorial & Gaussian)

$\therefore$ construct an unbiased estimator of the expectation, by sampling from $q_\theta(w \mid D)$

$$\begin{aligned} \mathbb{E}_q\left[p_{\mathcal{D}}\left(y^* \mid x^*\right)\right] & =\int q_\theta(w \mid \mathcal{D}) p_w(y \mid x) d w \\ & \approx \frac{1}{T} \sum_{t=1}^T p_{w_t}\left(y^* \mid x^*\right) \end{aligned} .$$

(T : pre-defined number of samples)

Predictive variance : $\operatorname{Var}_q\left(p\left(y^* \mid x^*\right)\right)=\mathbb{E}_q\left[y y^T\right]-\mathbb{E}_q[y] \mathbb{E}_q[y]^T$.

can be decomposed into **(1) aleatoric** and **(2) epistemic** uncertainty

$$\operatorname{Var}_q\left(p\left(y^* \mid x^*\right)\right)=\underbrace{\frac{1}{T} \sum_{t=1}^T \operatorname{diag}\left(\hat{p}_t\right)-\hat{p}_t \hat{p}_t^T}_{\text {aleatoric }}+\underbrace{\frac{1}{T} \sum_{t=1}^T\left(\hat{p}_t-\bar{p}\right)\left(\hat{p}_t-\bar{p}\right)^T}_{\text {epistemic }} .$$

- where $\bar{p}=\frac{1}{T} \sum_{t=1}^T \hat{p}_t$ and $\hat{p}_t=\operatorname{Softmax}\left(f_{w_t}\left(x^*\right)\right)$.

Since we can split into two parts as above...

we can see whether the quality of data is low **(high ALEATORIC uncertainty)**

or model itself is the cause of poor performance **(high EPISTEMIC uncertainty)**

5-3. Model Pruning

reduction in the model weights parameters!

- method 1) since parameter is doubled (for mean & var), reduce the number of filter into half
- method 2) L_1 norm
  - as most of the components will become close to 0

    non-zero components capture the most important features
  - make threshold! below that, make weight=0