[Paper review 8]

Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks

(Jose Miguel Hernandez-Lobato, Ryan P.Adams, 2015)

[Contents]

0. Abstract

Disadvantage of Backpropagation

- 1) have to tune LARGE NUMBER of HYPERPARAMETERS
- 2) lack of calibrated probabilistic predictions
- 3) tendency to overfit

Bayesian approach solve those problems!

But, Bayesian lack scalability to large dataset & network sizes

PBP (Probabilistic Backpropagation)

- scalable method for learning BNN
- forward propagation of probabilities

  backward computation of gradients
- provides accurate estimates of the posterior variance!

1. Introduction

NN solves wide range of supervised learning problems

success of NN is due to ability to train them on massive data (with stochastic optimization, backpropagation, ...)

How PBP solve those three problems (of original BP)?

- problem 1) have to tune LARGE NUMBER of HYPERPARAMETERS

  $\rightarrow$ automatically infer hyperparameter values (by marginalizing the out of the posterior)
- problem 2) lack of calibrated probabilistic predictions

  $\rightarrow$ account for uncertainty
- problem 3) tendency to overfit

  $\rightarrow$ average over parameter values (instead of single point), thus robust to overfitting!

Previous Bayesian Approach : lack of scalability

- ex 1) Laplace approximation (MacKay, 1992c)
- ex 2) Hamiltonian Monte Carlo (Neal, 1995)
- ex 3) Expected Propagation (Jylanki et al., 2014)
- ex 4) Variational Inference (Hinton & Camp, 1993)

Previous Bayesian Approach : has scalability, but.....

- ex 5) Scalable Variational Inference Appraoch (Graves, 2011)

  But, perform poorly in practice, due to noise from Monte Carlo approximations within the stochastic gradient computations
- ex 6) Scalable solution based on Expected Propagation (Soudry et al., 2014)

  (works with binary weights, but extension to continuous weights is unsatisfying)

  (does not produce estimates of posterior variance)

PBP : fast & dos not have the disadvantages of previous approaches!

2. Probabilistic Neural Network Models

data : $\mathcal{D}=\{\mathbf{x}_n, y_n\}_{n=1}^N,$ where $\mathbf{x}_n \in \mathbb{R}^D$, $y_n \in \mathbb{R}$,

probabilistic model : $y_n=f\left(\mathrm{x}_n ; \mathcal{W}\right)+\epsilon_n$

(+ additive noise variable : $\epsilon_n \sim \mathcal{N}\left(0, \gamma^{-1}\right)$)

Notation

L : number of layers

V_l : number of hidden units in layer l

$\mathcal{W}=\{\mathbf{W}_l\}_{l=1}^L:$ collection of $V_l \times\left(V_{l-1}+1\right)$ weight matrices

$\mathbf{a}_l=\mathbf{W}_l \mathbf{z}_{l-1} / \sqrt{V_{l-1}+1}:$ input to the l th layer (scaled)

$a(x)=\max (x, 0)$: ReLU activation function

(1) likelihood : $p(\mathbf{y} \mid \mathcal{W}, \mathbf{X}, \gamma)=\prod_{n=1}^{N} \mathcal{N}\left(y_{n} \mid f\left(\mathbf{x}_{n} ; \mathcal{W}\right), \gamma^{-1}\right)$

(2) prior : $p(\mathcal{W} \mid \lambda)=\prod_{l=1}^{L} \prod_{i=1}^{V_{l}} \prod_{j=1}^{V_{l-1}+1} \mathcal{N}\left(w_{ij,l} \mid 0, \lambda^{-1}\right)$

- Gaussian prior
- hyperprior for λ : $p(\lambda)=\operatorname{Gam}\left(\lambda \mid \alpha_{0}^{\lambda}, \beta_{0}^{\lambda}\right)$
- prior for noise precision γ : $p(\gamma)=\operatorname{Gam}\left(\gamma \mid \alpha_{0}^{\gamma}, \beta_{0}^{\gamma}\right)$

(3) posterior : $p(\mathcal{W}, \gamma, \lambda \mid \mathcal{D})=\frac{p(\mathbf{y} \mid \mathcal{W}, \mathbf{X}, \gamma) p(\mathcal{W} \mid \lambda) p(\lambda) p(\gamma)}{p(\mathbf{y} \mid \mathbf{X})}$

- normalizing constant : $p(\mathbf{y} \mid \mathbf{X})$

(4) predictive : $p\left(y_{\star} \mid \mathbf{x}_{\star}, \mathcal{D}\right)=\int p\left(y_{\star} \mid \mathbf{x}_{\star} \mathcal{W}, \gamma\right) p(\mathcal{W}, \gamma, \lambda \mid \mathcal{D}) d\gamma d\lambda d\mathcal{W}$

- where $p\left(y_{\star} \mid \mathbf{x}_{\star}, \mathcal{W}, \gamma\right)=\mathcal{N}\left(y_{\star} \mid f\left(\mathbf{x}_{\star}\right), \gamma\right)$
- $p(\mathcal{W}, \gamma, \lambda \mid \mathcal{D}) \text{ and } p\left(y_{\star} \mid \mathrm{x}_{\star}\right)$ is not tractable in most cases

  thus, use approximate inference

3. Probabilistic Backpropagation

2 phase of original BP :

- phase 1) propagate forward through the network to compute the function output & loss
- phase 2) derivatives of training loss (w.r.t weights) are propagated back

2 phase of PBP :

- do not use POINT estimates for the weights

  instead, use "collection of 1-D Gaussian" (each one approximating the marginal posterior distribution)
- phase 1) (same)
- phase 2)
  - weights are random $\rightarrow$ activations produced in each layer are also random $\rightarrow$ result in intractable distribution!

    sequentially approximates each of these distributions with a collection of 1-D Gaussian match their marginal mean & variance
  - instead of prediction error, use "logarithm of the marginal probability of the target variable"

    gradients of this quantity (w.r.t mean & variances) of the approximate Gaussian posterior are propagated back!

current prior : $q(w)=\$\mathcal{N}(w \mid m, v)$

updated prior : $s(w)=Z^{-1} f(w) \mathcal{N}(w \mid m, v)$

- Z : normalizing constant
- $s(w)$ have a complex form $\rightarrow$ approximate with simpler distribution (= use same form as q)

approximated upated prior : $q^{\text {new }}(w)=\mathcal{N}\left(w \mid m^{\text {new }}, v^{\text {new }}\right)$

- by minimizing KL-divergence between s and q^{new}
- $m^{\text {new }}=m+v \frac{\partial \log Z}{\partial m}$
- $v^{\text {new }}=v-v^{2}\left[\left(\frac{\partial \log Z}{\partial m}\right)^{2}-2 \frac{\partial \log Z}{\partial v}\right]$
- those two distributions (s and q^{new})have same mean & variance

Detailed description of PBP

- ADF (assumed density filtering) method
- uses some of the improvements on ADF given by expected propagation (Minka, 2001)

3-1. PBP as an ADF(Assumed Density Filtering) method

approximate the exact posterior of NN (with factored distribution)

$$q(\mathcal{W}, \gamma, \lambda)=\left[\prod_{l=1}^{L} \prod_{i=1}^{V_{l}} \prod_{j=1}^{V_{l-1}+1} \mathcal{N}\left(w_{i j, l} \mid m_{i j, l}, v_{i j, l}\right)\right] \times \operatorname{Gam}\left(\gamma \mid \alpha^{\gamma}, \beta^{\gamma}\right) \operatorname{Gam}\left(\lambda \mid \alpha^{\lambda}, \beta^{\lambda}\right)$$

approximation parameters are determind by ADF method

first, $q(\mathcal{W}, \gamma, \lambda)$ is initialized to uniform

- $m_{ij,l}=0, v_{ij,l}=\infty$
- $\alpha^{\gamma}=\alpha^{\lambda}=1$
- $\beta^{\gamma}=\beta^{\lambda}=0$

PBP iterates iver the factors in the numerator of $p(\mathcal{W}, \gamma, \lambda \mid \mathcal{D})=\frac{p(\mathbf{y} \mid \mathcal{W}, \mathbf{X}, \gamma) p(\mathcal{W} \mid \lambda) p(\lambda) p(\gamma)}{p(\mathbf{y} \mid \mathbf{X})}$

and sequentially incorporates each of these factors into the approximation in $q(\mathcal{W}, \gamma, \lambda)$

There are...

- 2 factors $\rightarrow$ for the priors on γ and λ ($p(\lambda)=\operatorname{Gam}\left(\lambda \mid \alpha_{0}^{\lambda}, \beta_{0}^{\lambda}\right)$, $p(\gamma)=\operatorname{Gam}\left(\gamma \mid \alpha_{0}^{\gamma}, \beta_{0}^{\gamma}\right)$)
- $\prod_{l=1}^{L} V_{l}\left(V_{l-1}+1\right)$ factors $\rightarrow$for the prior on W ($p(\mathcal{W} \mid \lambda)=\prod_{l=1}^{L} \prod_{i=1}^{V_{l}} \prod_{j=1}^{V_{l-1}+1} \mathcal{N}\left(w_{i j, l} \mid 0, \lambda^{-1}\right)$)
- N factors $\rightarrow$ for likelihood ($p(\mathbf{y} \mid \mathcal{W}, \mathbf{X}, \gamma)=\prod_{n=1}^{N} \mathcal{N}\left(y_{n} \mid f\left(\mathbf{x}_{n} ; \mathcal{W}\right), \gamma^{-1}\right)$)

3-2. Incorporating the PRIOR factors into q

priors on γ and λ

resulting update : $\alpha_{\text{new}}^{\gamma} = \alpha_{0}^{\gamma}, \beta_{\text{new}}^{\gamma} = \beta_{0}^{\gamma}, \alpha_{\text{new}}^{\lambda} = \alpha_{0}^{\lambda}$

- $\alpha_{\text{new}}^{\lambda} = \left[ZZ_2 Z_1^{-2}\left(\alpha^{\lambda}+1\right)/\alpha^{\lambda}-1.0\right]^{-1}$
- $\beta_{\text{new}}^{\lambda} = \left[Z_2 Z_1^{-1}\left(\alpha^{\lambda}+1\right)/\beta^{\lambda}-Z_1 Z^{-1}\alpha^{\lambda}/\beta^{\lambda}\right]^{-1}$

Notation

- Z : normalizer of s
- Z_1 : value of Z when α^{λ} is increased by 1 unit
- Z_2 : value of Z when α^{λ} is increased by 2 unit

How to find Z?

$$\begin{aligned}
Z= & \int \mathcal{N}\left(w_{ij,l} \mid 0, \lambda^{-1}\right) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma d\lambda \\
= & \int \mathcal{N}\left(w_{ij,l} \mid 0, \lambda^{-1}\right) \mathcal{N}\left(w_{ij,l} \mid m_{ij,l}, v_{ij,l}\right) \\
& \times \operatorname{Gam}\left(\lambda \mid \alpha^{\lambda}, \beta^{\lambda}\right) dw_{ij,l} d\lambda \\
= & \int \mathcal{T}\left(w_{ij,l} \mid 0, \beta^{\lambda}/\alpha^{\lambda}, 2\alpha^{\lambda}\right) \mathcal{N}\left(w_{ij,l} \mid m_{ij,l}, v_{ij,l}\right) dw_{ij,l} \\
\approx & \int \mathcal{N}\left(w_{ij,l} \mid 0, \beta^{\lambda}/\left(\alpha^{\lambda}-1\right)\right) \mathcal{N}\left(w_{ij,l} \mid m_{ij,l}, v_{ij,l}\right) dw_{ij,l} \\
= & \mathcal{N}\left(m_{ij,l} \mid 0, \beta^{\lambda}/\left(\alpha^{\lambda}-1\right)+v_{ij,l}\right)
\end{aligned}$$

where $\mathcal{T}(\cdot \mid \mu, \beta, \nu)$ denotes a Student's t distribution with mean μ, variance parameter β and degrees of freedom ν

approximate Student's t density with Gaussian density

3-3. Incorporating the LIKELIHOOD factors into q

N factors $\rightarrow$ for likelihood ($p(\mathbf{y} \mid \mathcal{W}, \mathbf{X}, \gamma)=\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid f\left(\mathbf{x}_n ; \mathcal{W}\right), \gamma^{-1}\right)$)

update for all the $m_{ij,1}$ and $v_{ij,l}$

assume an approximating Gaussian with mean m^{z_L} and variance v^{z_L}

How to find Z?

$$\begin{aligned}
Z &= \int \mathcal{N}\left(y_n \mid f\left(\mathbf{x}_n \mid \mathcal{W}\right), \gamma^{-1}\right) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma, d\lambda \\
&\approx \int \mathcal{N}\left(y_n \mid z_L, \gamma^{-1}\right) \mathcal{N}\left(z_L \mid m^{z_L}, v^{z_L}\right) \operatorname{Gam}(\gamma \mid \alpha^{\gamma}, \beta^{\gamma}) z_L d\gamma \\
&= \int \mathcal{T}\left(y_n \mid z_L, \beta^{\gamma}/\alpha^{\gamma}, 2\alpha^{\gamma}\right) \mathcal{N}\left(z_L \mid m^{z_L}, v^{z_L}\right) dz_L \\
&\approx \mathcal{N}\left(y_n \mid m^{z_L}, \beta^{\gamma}/\left(\alpha^{\gamma}-1\right)+v^{z_L}\right)
\end{aligned}$$

where $z_L=f\left(\mathbf{x}_i \mid \mathcal{W}\right) \sim \mathcal{N}\left(m^{z_L}, v^{z_L}\right)$

How to find (m^{z_L}, v^{z_L})

$\mathbf{a}_l = \mathbf{W}_l \mathbf{z}_{l-1} / \sqrt{V_{l-1}+1},$

- mean : $\mathbf{m}^{\mathbf{a}_l} = \mathbf{M}_l \mathbf{m}^{\mathbf{z}_{l-1}} / \sqrt{V_{l-1}+1}$
- variance : $\mathbf{v}^{\mathbf{a}_l} = \left[(\mathbf{M}_l \circ \mathbf{M}_l) \mathbf{v}^{\mathbf{z}_{l-1}} + \mathbf{V}_l (\mathbf{m}^{\mathbf{z}_{l-1}} \circ \mathbf{m}^{\mathbf{z}_{l-1}}) + \mathbf{V}_l \mathbf{v}^{\mathbf{z} l-1} \right] / (V_{l-1}+1)$

$\mathbf{b}_l = a(\mathbf{a}_l)$

- mean : $m_i^{\mathrm{b}_l} = \Phi(\alpha_i) v_i'$
- variance : $v_i^{\mathbf{b}_l} = m_i^{\mathbf{b}_l} v_i' \Phi(-\alpha_i) + \Phi(\alpha_i) v_i^{\mathbf{a}_l} (1 - \gamma_i (\gamma_i + \alpha_i))$

where $v_i' = m_i^{\mathrm{a}_l} + \sqrt{v_i^{\mathrm{a}_l}} \gamma_i, \quad \alpha_i = \frac{m_i^{\mathrm{a}l}}{\sqrt{v_i^{\mathrm{a}}}}, \quad \gamma_i = \frac{\phi(-\alpha_i)}{\Phi(\alpha_i)}$

and Φ and ϕ are respectively the cdf / pdf of standard Gaussian.

output of the l^{th} layer + bias 1 :

$\mathbf{m}^{\mathbf{z}_l} = \left[\mathbf{m}^{\mathbf{b}_l}; 1 \right], \quad \mathbf{v}^{\mathbf{z}_l} = \left[\mathbf{v}^{\mathbf{b}_l}; 0 \right]$

to compute mean & variance ($\mathbf{m}^{\mathbf{z}_l}$ & $\mathbf{v}^{\mathbf{z}_l}$) , initialize $\mathbf{m}^{\mathbf{z}_0}$ = $[x_i; 1]$ & $\mathbf{v}^{\mathbf{z}_0} = 0$

implement iteratively

until we obtain $m^{z_L} = m_1^{\mathbf{a}_L}$ & $v^{z_L} = v_1^{\mathbf{a}_L}$

3-4. Expectation Propagation

EP imporves ADF by iteratively incorporating "each factor multiple times"

- each factor is "removed" from the current posterior approximation, re-estimated, and re-incorporated
- disadvantage : have to keep in memory of all the approximate factors
- impossible with massive data

  $\rightarrow$ instead, incorporate these factors multiple times "without removing" them from the current approximation

  (but can lead to underestimation of variance parameters)