[Paper review 6]

Weight Uncertainty in Neural Networks (Charles Blundell, et.al , 2015)

[Contents]

0. Abstract

Bayes by Backprop

- New, Efficient "Backpropagation-compatible Algorithm" for learning a probability distribution on the weights of NN
- regularizes the weights by "minimizing a compression cost"

  (= ELBO, Variational Free Energy)
- comparable performance to dropout
- demonstrate how learnt uncertainty can be used to improve "generalization" in non-linear regression
- exploration-exploitation trade-off in reinforcement learning

1. Introduction

plain feedforward NN : prone to OVERFITTING

$\rightarrow$ by using variational Bayesian learning, introduce "Uncertainty in Weights"

"Bayes by Backprop" suggests 3 motivations for introducing uncertainty on weights

- 1) regularization on weights
- 2) richer representations & predictions from cheap model averaging
- 3) exploration in simple RL problems (ex. contextual bandits)

Previous works to prevent overfitting

- 1) early stopping
- 2) weight decay

- 3) dropout (Hinton et al., 2012)

Summary

- All weights are represented by "distribution" (not a single fixed points)
- Instead of learning single NN, BBB trains "an ENSEMBLE of networks"

  (each network has its weights drawn from a distribution)
- unlike other ensemble methods, only doubles the number of parameters! ($\mu \& \sigma$)
- gradients can be made UNBIASED and can also be used with non-Gaussian priors!
- uncertainty in hidden unit $\rightarrow$ uncertainty about particular observation $\rightarrow$ regularization of the weights

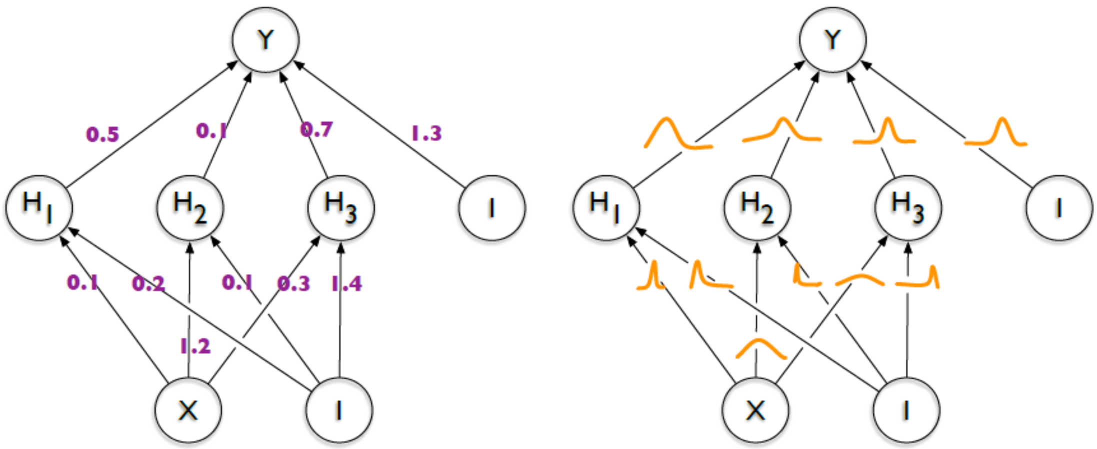

Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

2. Point Estimates of Neural Networks

probabilistic model : $P(y \mid x, w)$

- for categorical dist'n : cross-entropy, softmax loss
- for continuous dist'n : squared loss

Weights can be learnt by...

- MLE (Maximum Likelihood Estimator) :

$$\begin{aligned} \mathbf{w}^{\mathrm{MLE}} &= \arg\max_{\mathbf{w}} \log P(\mathcal{D} \mid \mathbf{w}) \\ &= \arg\max_{\mathbf{w}} \sum_i \log P\left(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{w}\right) \end{aligned}$$

- MAP (Maximum a Posteriori) : can introduce REGULARIZATION :

$$\begin{aligned} \mathbf{w}^{\mathrm{MAP}} &= \arg\max_{\mathbf{w}} \log P(\mathbf{w} \mid \mathcal{D}) \\ &= \arg\max_{\mathbf{w}} \log P(\mathcal{D} \mid \mathbf{w}) + \log P(\mathbf{w}) \end{aligned}$$

if $\mathbf{w}$ is a Gaussian prior : $L2$ regularization

if $\mathbf{w}$ is a Laplace prior : $L1$ regularization

3. Being Bayesian by Backpropagation

Bayesian inference for neural networks calculates the posterior distribution $P(\mathbf{w} \mid \mathcal{D})$

predictive distribution : $P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}) = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}, \mathbf{w})]$

"Taking an expectation under the posterior distributions on weights = ensemble of uncountably infinite number of NN"

Variational Inference

- find $q(\mathbf{w} \mid \theta)$ that minimizes KL divergence

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

- cost function : "Variational Free energy" (= maximize ELBO)

$$\mathcal{F}(\mathcal{D}, \theta) = \mathrm{KL}[q(\mathbf{w} \mid \theta) \| P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D} \mid \mathbf{w})]$$

3-1. Unbiased Monte Carlo gradients

Reparameterzation trick :

deterministic function $t(\theta, \epsilon)$ transforms a sample of parameter-free noise ϵ & parameter θ into a sample from the variational posterior!

$\frac{\partial}{\partial \theta} \mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w}, \theta)] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \theta} + \frac{\partial f(\mathbf{w},\theta)}{\partial \theta}\right]$

Proof)

$$\begin{aligned}
\frac{\partial}{\partial \theta} \mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w}, \theta)] &= \frac{\partial}{\partial \theta} \int f(\mathbf{w}, \theta) q(\mathbf{w} \mid \theta) \mathrm{d}\mathbf{w} \\
&= \frac{\partial}{\partial \theta} \int f(\mathbf{w}, \theta) q(\epsilon) \mathrm{d}\epsilon \\
&= \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \theta} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \theta}\right]
\end{aligned}$$

using the trick above, approximate

- $\mathcal{F}(\mathcal{D}, \theta) = \mathrm{KL}[q(\mathbf{w} \mid \theta) \| P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D} \mid \mathbf{w})]$ as
  $\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^n \log q\left(\mathbf{w}^{(i)} \mid \theta\right) - \log P\left(\mathbf{w}^{(i)}\right) - \log P\left(\mathcal{D} \mid \mathbf{w}^{(i)}\right)$
- where $w^{(i)}$ denotes the i^{th} MC sample drawn from variational posterior $q(\mathbf{w}^{(i)} \mid \theta)$

found that a prior without an easy-to-compute closed form complexity cost performed the best

3-2. Gaussian Variational Posterior

suppose variational posterior = "diagonal Gaussian"

parameter : $\theta = (\mu, \rho)$ where $\sigma = \log\left(1 + \exp(\rho)\right)$

weight : $\mathbf{w} = t(\theta, \epsilon) = \mu + \log(1 + \exp(\rho)) \circ \epsilon$

Each step of optimization :

1. Sample $\epsilon \sim \mathcal{N}(0, I)$.
2. Let $\mathbf{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon$.
3. Let $\theta = (\mu, \rho)$.
4. Let $f(\mathbf{w}, \theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(\mathcal{D}|\mathbf{w})$.
5. Calculate the gradient with respect to the mean

$$\Delta_\mu = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \mu}. \tag{3}$$

6. Calculate the gradient with respect to the standard deviation parameter ρ

$$\Delta_\rho = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} \frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \rho}. \tag{4}$$

7. Update the variational parameters:

$$\mu \leftarrow \mu - \alpha \Delta_\mu \tag{5}$$
$$\rho \leftarrow \rho - \alpha \Delta_\rho. \tag{6}$$

$\frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}}$: shared for mean & variance

- also, excatly the same gradients find in plain backprop!

4. Key point

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

use $f(\mathbf{w}, \theta) = \log q(\mathbf{w} \mid \theta) - \log p(\mathbf{w}) - \log p(D \mid \mathbf{w})$ for training $q(\mathbf{w} \mid \theta)$