Bayes by Backprop Report

Dogancan Kebude

December 2018

1 Introduction

This is a short report on implementation of *Bayes by Backprop* regression introduced in [1]. The report will not contain the details explained in the paper but instead focus on how the distributions are chosen for variational inference, how the parameters are initialized and how the backpropagation is conducted.

The rest of the report is structured as follows: Sect. 2 will talk about the regression problems we are trying to solve, Sect. 3 will summarize the algorithmic flow, Sect. 4 will analyze the loss function and how the distributions are chosen, Sect. 5 will explain the gradient calculation process, Sect. 6 will describe how the parameters are updated, Sect. 7 will describe the parameter initialization and finally, Sect. 7 will discuss the observations and conclude the report.

2 Regression Problems

In this implementation, we are looking for a solution to two toy regression problems:

The first one is the regression problem explained in Sect. 5.2 of [1]. The data is sampled from: $y = x + (0.3sin(2\pi(x + \epsilon))) + (0.3sin(4\pi(x + \epsilon))) + \epsilon$, where $\epsilon \sim N(0, 0.02)$.

The second one is the toy regression problem explained in Sect. 3.3. of [2]. The data is sampled from $y = x^3 + \epsilon$, where $\epsilon \sim N(0, 9)$.

Both problems were solved with the model explained in Sect. 3.2 of [1]. However, the discussion section only contains the results for the regression problem taken from [2].

3 Algorithmic Flow

The reader should note the algorithmic flow to have a better grasp over the derivations. For every iteration (i):

$$\epsilon \sim N(0, 1) \tag{1}$$

$$\theta = (\mu, \rho) \tag{2}$$

$$\sigma = \log{(1 + e^{\rho})} \tag{3}$$

$$\mathbf{w}^{(i)} = \mu + \sigma \circ \epsilon \tag{4}$$

The above steps lead on to the standard neural network calculations and finally we calculate the loss and backpropagate to update the terms μ and θ.

4 Loss Function

As it is provided in [1], Eqn. 2, the loss function for the network is an approximation of variational free energy, a.k.a. expected lower bound (ELBO).

$$F(D, \theta) = \sum_{i=1}^{n} \log q(\mathbf{w}^{(i)}|\theta) - \log P(\mathbf{w}^{(i)}) - \log P(D|\mathbf{w}^{(i)}) \tag{5}$$

This implies a loss function that can be divided into three distributions. The implemented version chooses all three to be Gaussian. The sections below solely consist of mathematical derivations of functions to implement in code.

4.1 Variational Posterior

$$\log q(\mathbf{w}^{(i)}|\theta) = \log{(\frac{1}{\sigma^{(i)}\sqrt{2\pi}} e^{-\frac{(\mathbf{w}^{(i)} - \mu^{(i)})^2}{2(\sigma^{(i)})^2}})} \tag{6}$$

$$\log q(\mathbf{w}^{(i)}|\theta) = \log 1 - \log \sigma^{(i)} - \frac{1}{2} \log 2\pi - \frac{(\mathbf{w}^{(i)} - \mu^{(i)})^2}{2(\sigma^{(i)})^2} \tag{7}$$

$$\log q(\mathbf{w}^{(i)}|\theta) = \log 1 - \log \sigma^{(i)} - \frac{1}{2} \log 2\pi - \frac{(\mathbf{w}^{(i)} - \mu^{(i)})^2}{2(\sigma^{(i)})^2} \tag{8}$$

At this point we can drop the term with 2π as it will cause the same amount of loss on all samples.

$$\log q(\mathbf{w}^{(i)}|\theta) = - \log \sigma^{(i)} - \frac{(\mathbf{w}^{(i)} - \mu^{(i)})^2}{2(\sigma^{(i)})^2} \tag{9}$$

Utilizing Eqn. 4, we get:

$$\log q(\mathbf{w}^{(i)}|\theta) = - \log \sigma^{(i)} - \frac{1}{2}\epsilon^2 \tag{10}$$

4.2 Prior

Gaussian prior for $\mathbf{w}^{(i)}$ is chosen to be zero mean and its standard deviation is chosen to be the same as the data's.

$$\log P(\mathbf{w}^{(i)}) = \log\left(\frac{1}{\sigma_{data}\sqrt{2\pi}}e^{-\frac{(\mathbf{w}^{(i)})^2}{2\sigma_{data}^2}}\right) \tag{11}$$

$$\log P(\mathbf{w}^{(i)}) = \log 1 - \log\sigma_{data} - \frac{1}{2}\log 2\pi - \frac{(\mathbf{w}^{(i)})^2}{2\sigma_{data}^2} \tag{12}$$

Where σ_{data} is the data's standard deviation. We can again drop the 2π term and conclude the prior to be:

$$\log P(\mathbf{w}^{(i)}) = -\log\sigma_{data} - \frac{1}{2}\frac{(\mathbf{w}^{(i)})^2}{\sigma_{data}^2} \tag{13}$$

4.3 Likelihood

The log likelihood is chosen to be Gaussian with zero mean and unit variance:

$$\log P(D|\mathbf{w}^{(i)}) = \log\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{(y-\hat{y})^2}{2}}\right) \tag{14}$$

$$\log P(D|\mathbf{w}^{(i)}) = \log 1 - \frac{1}{2}\log 2\pi - \frac{(y-\hat{y})^2}{2} \tag{15}$$

Dropping the 2π term:

$$\log P(D|\mathbf{w}^{(i)}) = -\frac{1}{2}(y-\hat{y})^2 \tag{16}$$

4.4 Final Loss Function

The three subsections above lead our loss function for regression to become:

$$F(D,\theta) = \sum_{i=1}^{n} -\log\sigma^{(i)} - \frac{1}{2}\epsilon^2 + \frac{1}{2}(\mathbf{w}^{(i)})^2 + \frac{1}{2}(y-\hat{y})^2 \tag{17}$$

For any $D^{(i)}$ and $\theta^{(i)}$ pair, the episodic loss function is, hence:

$$F(D^{(i)},\theta^{(i)}) = \frac{1}{n}[-\log\sigma^{(i)} - \log\sigma_{data} - \frac{1}{2}\epsilon^2 + \frac{1}{2}\frac{(\mathbf{w}^{(i)})^2}{\sigma_{data}^2}] + \frac{1}{2}(y-\hat{y})^2 \tag{18}$$

5 Gradients

The gradients should be calculated for μ and ρ parameters for optimization through gradient descent.

5.1 Gradient with respect to μ

Since variational posterior terms in Eqn. 18 do not contain a μ, their partial derivatives are not shown below.

$$\frac{\partial F(D^{(i)},\theta^{(i)})}{\partial \mu^{(i)}} = \frac{1}{n}\frac{\mathbf{w}^{(i)}}{\sigma_{data}^{2}} + \frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}}\frac{\partial \mathbf{w}^{(i)}}{\partial \mu^{(i)}} \tag{19}$$

Where $\frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}}$ is the standard backpropagation through the neural network. The network gradients will be shown later.

Also, as can be seen from Eqn. 4:

$$\frac{\partial \mathbf{w}^{(i)}}{\partial \mu^{(i)}} = 1 \tag{20}$$

$$\frac{\partial F(D^{(i)},\theta^{(i)})}{\partial \mu^{(i)}} = \frac{1}{n}\frac{\mathbf{w}^{(i)}}{\sigma_{data}^{2}} + \frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}} \tag{21}$$

5.2 Gradient with respect to ρ

All terms of Eqn. 18 except ϵ contain a ρ element inside and hence will be handled one by one for a clearer view:

5.2.1 The $\sigma^{(i)}$ Term

$$\frac{\partial[\log \sigma^{(i)}]}{\partial \rho^{(i)}} = \frac{\partial[\log \sigma^{(i)}]}{\partial \sigma^{(i)}}\frac{\partial \sigma^{(i)}}{\partial u^{(i)}}\frac{\partial u^{(i)}}{\partial \rho^{(i)}} \tag{22}$$

where $u^{(i)} = 1 + e^{(\rho)(i))}$

$$\frac{\partial[\log \sigma^{(i)}]}{\partial \rho^{(i)}} = \frac{1}{\sigma^{(i)}}\frac{1}{1+e^{(\rho^{(i)})}}e^{(\rho^{(i)})} \tag{23}$$

$$\frac{e^{(\rho^{(i)})}}{1+e^{(\rho^{(i)})}} = \frac{1}{\frac{1+e^{(\rho^{(i)})}}{e^{(\rho^{(i)})}}} = \frac{1}{1+e^{(-\rho^{(i)})}} \tag{24}$$

$$\frac{\partial[\log \sigma^{(i)}]}{\partial \rho^{(i)}} = \frac{1}{\sigma^{(i)}}\frac{1}{1+e^{(-\rho^{(i)})}} \tag{25}$$

5.2.2 The $\mathbf{w}^{(i)}$ Term

$$\frac{\partial[\frac{1}{2}\frac{(\mathbf{w}^{(i)})^{2}}{\sigma_{data}^{2}}]}{\partial \rho^{(i)}} = \frac{\mathbf{w}^{(i)}}{\sigma_{data}^{2}}\frac{\partial \mathbf{w}^{(i)}}{\partial \sigma^{(i)}})\frac{\partial \sigma^{(i)}}{\partial \rho^{(i)}} = \frac{\mathbf{w}^{(i)}}{\sigma_{data}^{2}}\frac{\epsilon}{1+e^{(-\rho^{(i)})}} \tag{26}$$

5.2.3 The Likelihood Term

$$\frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \rho^{(i)}} = \frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}} \frac{\partial \mathbf{w}^{(i)}}{\partial \rho^{(i)}} \tag{27}$$

As it can be seen from Eqn. 26:

$$\frac{\partial \mathbf{w}^{(i)}}{\partial \rho^{(i)}} = \frac{\epsilon}{1 + e^{(-\rho^{(i)})}} \tag{28}$$

$$\frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \rho^{(i)}} = \frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}} \frac{\epsilon}{1 + e^{(-\rho^{(i)})}} \tag{29}$$

5.2.4 Complete gradient with respect to ρ

$$\frac{\partial F(D^{(i)}, \theta^{(i)})}{\partial \rho^{(i)}} = \frac{1}{n}[(-\frac{1}{\sigma^{(i)}} + \frac{\mathbf{w}^{(i)}}{\sigma^2_{data}}\epsilon)\frac{1}{1 + e^{(-\rho^{(i)})}}] + \frac{\partial P(D^{(i)}|\mathbf{w}^{(i)})}{\partial \mathbf{w}^{(i)}} \frac{\epsilon}{1 + e^{(-\rho^{(i)})}} \tag{30}$$

5.3 Network Backpropagation

The forward pass of the utilized network utilizes ReLU as the activation function, hence the forward pass of the network can be described as:

$$x_{train} * \mathbf{w}_0 + b_0 = z_0 \tag{31}$$

$$ReLU(z_0) = a_0 \tag{32}$$

$$a_0 * \mathbf{w}_1 + b_1 = z_1 \tag{33}$$

$$ReLU(z_1) = a_1 \tag{34}$$

$$...$$

$$ReLU(z_{n-1}) = a_{n-1} \tag{35}$$

$$a_{n-1} * \mathbf{w}_n + b_n = \hat{y} \tag{36}$$

Hence, the backward pass looks as follows:

$$\frac{\partial P(D^{(i)}|\mathbf{b}_n^{(i)})}{\partial \mathbf{b}_n^{(i)}} = -(y - \hat{y}) \tag{37}$$

$$\frac{\partial P(D^{(i)}|\mathbf{w}_n^{(i)})}{\partial \mathbf{w}_n^{(i)}} = -(y - \hat{y})a_{n-1} \tag{38}$$

$$\frac{\partial P(D^{(i)}|\mathbf{b}_{n-1}^{(i)})}{\partial \mathbf{b}_{n-1}^{(i)}} = -(y - \hat{y})\mathbf{w}_n^{(i)}[z_{n-1} > 0] \tag{39}$$

$$\frac{\partial P(D^{(i)}|\mathbf{w}_{n-1}^{(i)})}{\partial \mathbf{w}_{n-1}^{(i)}} = -(y - \hat{y})\mathbf{w}_n^{(i)}[z_{n-1} > 0]a_{n-2} \tag{40}$$

$$...$$

$$\frac{\partial P(D^{(i)}|\mathbf{b}_0^{(i)})}{\partial \mathbf{b}_0^{(i)}} = -(y - \hat{y})\mathbf{w}_n^{(i)}[z_{n-1} > 0]\mathbf{w}_{n-1}^{(i)}[z_{n-2} > 0]...\mathbf{w}_1^{(i)}[z_0 > 0] \tag{41}$$

$$\frac{\partial P(D^{(i)}|\mathbf{w}_0^{(i)})}{\partial \mathbf{w}_0^{(i)}} = -(y - \hat{y})\mathbf{w}_n^{(i)}[z_{n-1} > 0]\mathbf{w}_{n-1}^{(i)}[z_{n-2} > 0]...\mathbf{w}_1^{(i)}[z_0 > 0]x_{train} \tag{42}$$

Since the μ and ρ parameters are defined for all weights and biases, the gradients should be calculated for all weights and all μ, ρ pairs.

6 Update

Following the algorithmic flow in Sect. 2, we end up with calculating the loss and then gradients to update μ and ρ pairs at each iteration as follows:

$$\mu = \mu - \eta \frac{\partial F(D^{(i)}, \theta^{(i)})}{\partial \mu^{(i)}} \tag{43}$$

$$\rho = \rho - \eta \frac{\partial F(D^{(i)}, \theta^{(i)})}{\partial \rho^{(i)}} \tag{44}$$

Where η is the learning rate.

7 Parameter Initialization

The initial μ parameters are sampled from zero mean Gaussians with 0.01 variance and the initial ρ parameters are sampled from uniform distributions between -0.01 and 0.01. As $\sigma = \log(1 + e^{\rho})$, if we choose ρ close to zero, standard deviation of the weights and biases will be too large, causing the distributions to be too wide, i.e. $log(1 + e^{-0.01} = 0.7$ and $log(1 + e^{0.01} = 0.7$. Hence, we use a small negative offset (-5) to start ρ parameters off: $log(1 + e^{-5}) = 0.007$, improving the learning process.

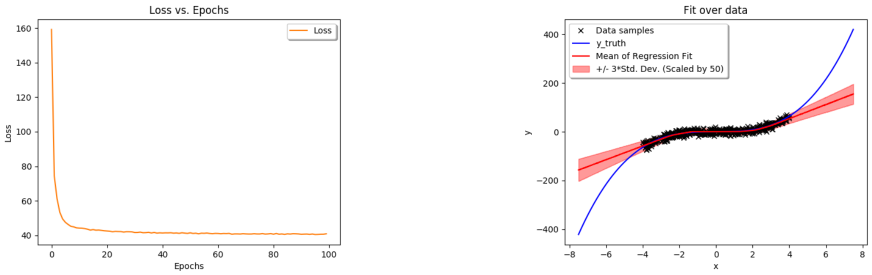
8 Discussion

The *Bayes by Backprop* implementation can successfully train over different data samples and different regression problems. The variational inference helps with the overfitting problem with the help of wider weight distributions (i.e. higher standard deviation) where we do not have enough data, and narrow weight distributions where we have data samples (Fig. 1). If we sample the data from a narrower range (i.e. $x = [-2.5, 2.5]$), then we can see a smaller variation increase between $(-\infty, -2.5)$ and between $(2.5, \infty)$ (Fig. 2). This is also corroborated through Fig. 3: As the data is sampled from a narrower range (i.e. $x = [-1, 1]$), the linearity of the fit and the model's belief of its correctness increases. Note that, the sample counts are decreased between executions to keep the sample number per area equivalent between all three cases.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1613–1622. JMLR.org, 2015.

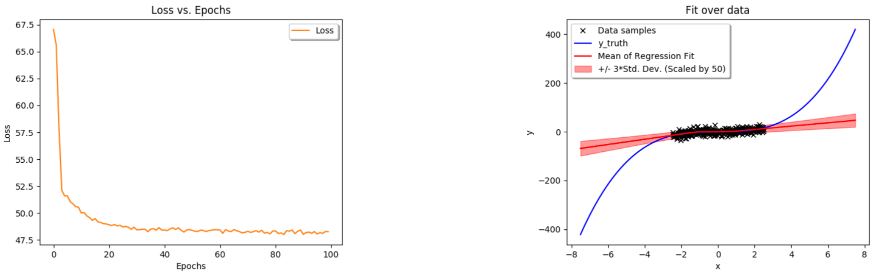
[2] Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy, and Max Welling. Bayesian dark knowledge. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3438–3446. Curran Associates, Inc., 2015.



(a) Loss vs. Epochs

(b) Regression fit over samples

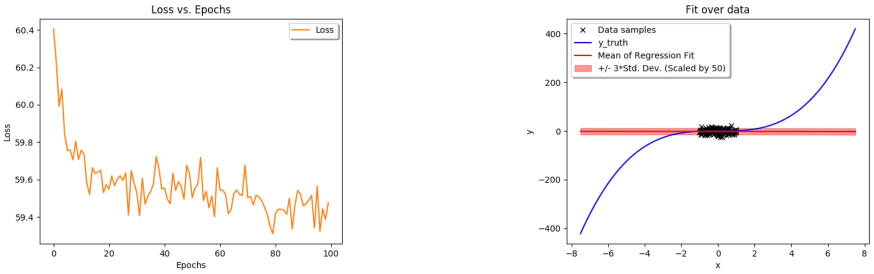
Figure 1: The training results for two layer network with hidden layer size 100, $1e-3$ learning rate, sample count 500 and 100 epochs. The data is sampled between $[-4, 4]$



(a) Loss vs. Epochs

(b) Regression fit over samples

Figure 2: The training results for two layer network with hidden layer size 100, $1e-3$ learning rate, sample count 300 and 100 epochs. The data is sampled between $[-2.5, 2.5]$



(a) Loss vs. Epochs

(b) Regression fit over samples

Figure 3: The training results for two layer network with hidden layer size 100, $1e-3$ learning rate, sample count 125 and 100 epochs. The data is sampled between $[-1, 1]$