0.1 Explaining approximation of parameter posterior

0.1.1 Introduction

This document accompanies my blog post on robromijnders.github.io about Bayesian deep learning.

The goal of this document is to explain the objective function for our approximation. The blog post motivates why we want to approximate the parameter posterior. It also provides an objective function and some python pseudo code to implement it. For those who want to see the derivation: I wrote this document for you!

0.1.2 Variational inference

So Bayes rule gives us a posterior over our parameters and we want to approximate it. This approximation goes by the name *variational inference.* To repeat from the blog post, our posterior is:

$$\begin{aligned}
posterior &\propto likelihood \times prior && (1)\\
p(w|data) &\propto p(data|w) \times p(w) && (2)\\
& && (3)\\
log\ posterior &= log\ likelihood + log\ prior + constant && (4)\\
&= -classification\ loss - \lambda \sum_i w_i^2 + constant && (5)
\end{aligned}$$

So far nothing new.

Now for our approximation, we need to quantify how good any approximation is. By convention of variational inference, we choose the KL divergence. The KL divergence measures the *divergence* between any two distributions. Let us name our approximation $q(w)$. Then our objective for the approximation is:

$$D_{KL}(q(w)||p(w|data))$$

If you ever wondered why we take the KL divergence from q to p instead of the other way around: Murphy explained that in his book *Machine learning: a probabilistic perspective. Section 21.2.2.*

Get ready, let's write out the math!

$$\begin{aligned}
D_{KL}(q(w)||p(w|data)) &= E_{q(w)}[\log q(w) - \log p(w|data)] + constant && (6)\\
&= E_{q(w)}[-\log p(data|w)] + D_{KL}(q(w)||p(w)) && (7)
\end{aligned}$$

So this last line has two terms

- **The first term** is the classification loss, under the expectation of our approximation. What does that practically mean? It means that we optimize the approximation one sample at a time: we sample one weight, do a gradient descent step and repeat.

- **The second term** is the KL divergence between our prior and approximation. In our blog post we motivated that we will approximate the parameter posterior with a Gaussian. So the second term actually refers to the KL divergence between two Gaussians, for which we know the formula!

With these two remarks, we can simplify our equation to:

$$D_{KL}(q(w)||p(w|data)) = classificationloss + \log\frac{\sigma_{prior}}{\sigma} + \frac{\sigma^2 + (\mu - \mu_{prior})^2}{2\sigma^2_{prior}} + constant$$

Note that in this simplification, we drop the expectation over our approximation. The classification loss should be read under randomly sampled parameters from our approximation.

Now we make assumptions on our prior and simplify the equation even further. For the parameters of a neural network, we assume a priori that:

- the parameters have zero mean, i.e. we have no information to set the parameters to a specific value.
- the parameters have a variance $\frac{1}{\lambda}$. We use this λ to conform with our familiar equation 5

.

These assumptions simplify our equation to:

$$D_{KL}(q(w)||p(w|data)) = classificationloss + \sum_i -\log\sigma_i + \frac{1}{2}\lambda\sigma^2 + \frac{1}{2}\lambda\mu^2 + constant \quad (8)$$

0.1.3 Interpretation

Equations 5 and 8 look remarkably similar. Let's repeat them:

$$\begin{aligned} -\log posterior = classification\ loss + &\quad \lambda\sum_i w_i^2 + & constant \\ D_{KL}(q(w)||p(w|data)) = classificationloss + &\quad \sum_i \frac{1}{2}\lambda\mu^2 - \log\sigma_i + \frac{1}{2}\lambda\sigma^2 + & constant \end{aligned}$$

- Both equations have the classification loss
- The w_i^2 term in the first equation and the μ^2 are similar
- The difference comes in the $\frac{1}{2}\lambda\sigma^2 - \log\sigma$ term. This term *pulls* the standard deviations to the prior standard deviation. In Figure 1 we plot an example of this term for $\lambda = 1$

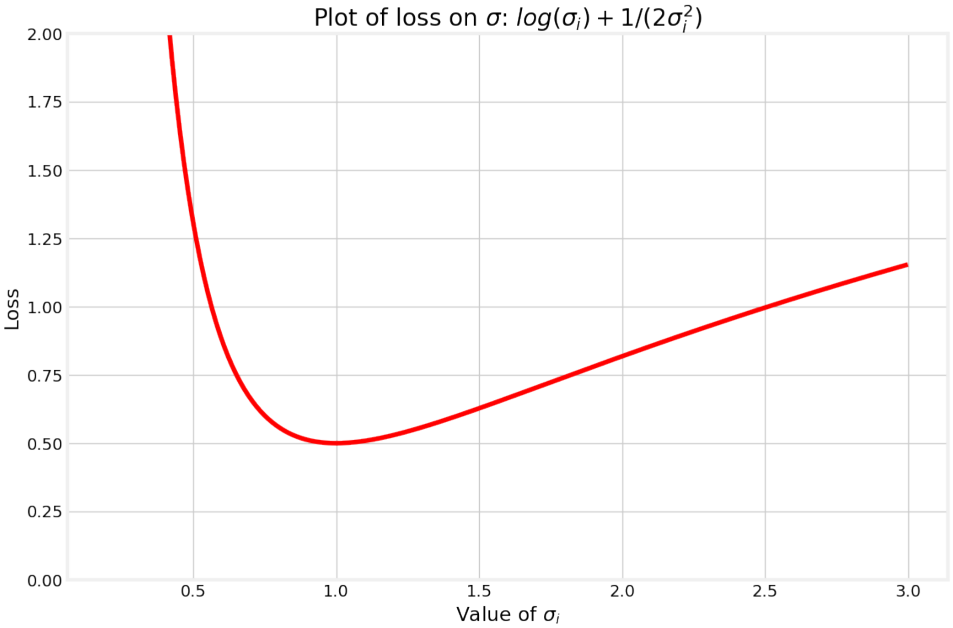

Figure 1: Plot of the $\frac{1}{2}\lambda\sigma^2 - \log\sigma$ term for $\lambda = 1$