Introduction

Issues:

- Existing variational inference (VI) methods, e.g., black-box VI (BBVI) [2], require significant changes to existing deep-learning implementation and demand large memory.

Contributions:

- For Gaussian variational approximations, we propose a method called Vprop which can be implemented with a slight modification to RMSprop code.
- Vprop simplifies the implementation of BBVI and also reduces its memory requirement by half.

- Vprop is a natural-gradient method for VI and also related to Newton’s method.

Optimization Algorithms & VI

Maximum Likelihood Estimation (MLE):

We can perform MLE by minimizing \( f(\theta) = -\log p(y|X, \theta) \) using RMSprop [1], as shown in Algorithm 2:

\[
\text{RMSprop:} \quad \begin{align*}
\theta_{t+1} &= \theta_t - \alpha_t (s_t + \delta t)^{-\gamma} \nabla f(\theta_t), \\
\mathbf{s}_{t+1} &= (1 - \beta_2) \mathbf{s}_t + \beta_2 (\nabla f(\theta_t))^2,
\end{align*}
\]

where \( s \) is the adaptive step-size scaling vector.

Gaussian Variational Inference:

In contrast, in Gaussian VI, we approximate the posterior distribution with a Gaussian distribution \( q(\theta) = \mathcal{N}(\mu, \sigma^2) \) by maximizing a variational lower-bound:

\[
\log p(y|X, \theta) = \max_{\mu, \sigma} \mathbb{E}_q[\log p(y|X, \theta) - \log q(\theta)] := \mathcal{L}(\mu, \sigma)
\]

where \( \phi(\theta) := \mathcal{N}(\theta|0, 1) \) is the Gaussian prior.

Black-box Variational Inference:

Black-box VI (BBVI) [2] optimizes \( \mu \) and \( \sigma \) by the following stochastic gradient-descent algorithm (or adaptive-gradient variants of it):

\[
\begin{align*}
\text{BBVI:} \quad &\mu_{t+1} = \mu_t + \gamma_t \nabla \mathcal{L}(\mu_t), \\
&\sigma_{t+1} = \sigma_t + \gamma_t \nabla \mathcal{L}(\sigma_t),
\end{align*}
\]

However, BBVI update has two issues:

- Memory: The number of parameters is doubled for adaptive-gradient since it needs to store scaling vectors for \( \mu \) and \( \sigma \).
- Implementation: Computing the gradients w.r.t. \( \sigma \) requires a different implementation from that of existing deep learning.

Conjugate-computation Variational Inference:

We propose Vprop to solve these issues by using a natural gradient method called conjugate-computation VI (CVI) [3]:

\[
\begin{align*}
\text{CVI:} \quad &\mu_{t+1} = \mu_t + \beta_t \nabla \mathcal{L}(\mu_t), \\
&\sigma_{t+1}^2 = \sigma_t^2 - 2\beta_t \nabla \mathcal{L}(\sigma_t).
\end{align*}
\]

Vprop

Derivation of Vprop:

Vprop is derived from the CVI update in two steps.

- Apply Bonnet’s and Price’s theorem:

\[
\nabla \mathcal{L}_\mu = -\mathbb{E}_q[\nabla \mathcal{L}(\theta)] - \lambda \mathbb{E}_q[\nabla \mathcal{L}(\theta)],
\]

\[
\nabla \mathcal{L}_\sigma = -\frac{1}{2} \mathbb{E}_q[(\nabla \mathcal{L}(\theta))^2] - \frac{1}{2} \mathbb{E}_q[\nabla \mathcal{L}(\theta)]^2.
\]

The Hessian is approximated by a Gauss-Newton approximation.

By defining \( \mathbf{s}_t := \sigma_t^2 \lambda \) and using the above two steps in the CVI update, we obtain Vprop:

\[
\text{Vprop-1:} \quad \begin{align*}
\mu_{t+1} &= \mu_t - \beta_t \left( \mathbf{s}_t + \lambda \right)^{-1} \mathbb{E}_q[\nabla \mathcal{L}(\theta)] + \lambda \mu_t, \\
\mathbf{s}_{t+1} &= (1 - \beta_2) \mathbf{s}_t + \beta_2 (\nabla \mathcal{L}(\theta))^2.
\end{align*}
\]

The expectations can be approximated by using one Monte-Carlo (MC) sample giving us Vprop-1:

\[
\text{Vprop-1:} \quad \begin{align*}
\theta_{t+1} &\sim \mathcal{N}(\mu_{t+1}, \sigma_{t+1}^2), \\
\mu_{t+1} &= \mu_t - \beta_t \left( \mathbf{s}_t + \lambda \right)^{-1} \mathbb{E}_q[\mathbf{s}_t \nabla \mathcal{L}(\theta)] + \lambda \mu_t, \\
\mathbf{s}_{t+1} &= (1 - \beta_2) \mathbf{s}_t + \beta_2 (\nabla \mathcal{L}(\theta))^2.
\end{align*}
\]

Comparison between Vprop and RMSprop:

Vprop-1 and RMSprop are similar but have 3 significant differences:

- Vprop-1 computes the gradients at \( \theta \) sampled from \( \mathcal{N}(\theta, \mathbf{s}_t^2) \) but RMSprop computes gradients at \( \theta = \mu_t \).
- RMSprop raise the scaling vector to a power of \( \frac{1}{2} \), while Vprop-1 does not.
- Vprop-1 adds \( \lambda \) to the gradient of \( \mu \) update.

Connections to natural gradient and Newton’s method:

For full Gaussian \( q(\theta) \), using similar derivations result in a variant of Newton’s method. The resulting update also resembles an online natural gradient descent proposed by Ollivier 2017 [4].

Experiments

Logistic regression

Multilayer-perceptron

References