Variational Inference for Monte Carlo Objectives

Andriy Mnih, Danilo J. Rezende

Google DeepMind

Overview

Insufficiently expressive variational posteriors prevent latent variable models from using their full capacity. Replacing the classical single-sample variational objective with its multi-sample counterpart has been shown to help when training models with continuous latent variables (Burda et al., 2016). We extend the approach by constructing an effective unbiased low-variance gradient estimator for multi-sample objectives, which can handle both discrete and continuous latent variables.

Monte Carlo objectives

The classical variational lower bound on $\log P_\theta(x)$:

$$\mathcal{L}(x) = E_{Q(h|x)}\left[\log \frac{P(x,h)}{Q(h|x)}\right] = \log P(x) + E_{Q(h|x)}\left[\log \frac{P(h|x)}{Q(h|x)}\right].$$

A tighter, multi-sample generalization of it (IWAE, Burda et al., 2016):

$$\mathcal{L}^{IWAE}(x) = E_{Q(h|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K} \frac{P(x,h^k)}{Q(h^k|x)}\right]$$

We develop a general low-variance gradient estimator for objectives of the form

$$\mathcal{L}^K(x) = E_{Q(h^{1:K}|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K} f(x,h^k)\right].$$

Gradient of the objective

Defining $\hat{l}(h^{1:K}) \equiv \frac{1}{K}\sum_{k=1}^{K} f(x,h^k)$, the gradient of $\mathcal{L}^K(x)$ can be written as

$$\frac{\partial}{\partial\theta}\mathcal{L}^K(x) = E_{Q(h^{1:K}|x)}\left[\sum_j \log \hat{l}(h^{1:K}) \frac{\partial}{\partial\theta}\log Q(h^j|x)\right] +$$

$$E_{Q(h^{1:K}|x)}\left[\sum_j \tilde{w}^j \frac{\partial}{\partial\theta}\log f(x,h^j)\right]$$

where $\tilde{w}^j \equiv \frac{f(x,h^j)}{\sum_{k=1}^{K} f(x,h^k)}$.

We can think of $\log \hat{l}(h^{1:K})$ as the learning signal for $Q(h^j|x)$, since it modulates the gradient of $\log Q(h^j|x)$. Estimating the first expectation using sampling is hard because this learning signal

i. can be arbitrarily negative and
ii. is the same for all K samples $h^1, ..., h^K$ (no credit assignment).

NVIL gradient estimator

Neural Variational Inference and Learning (NVIL, Mnih & Gregor, 2014) addresses (i) by learning an input-dependent predictor $b(x)$ of $\log \hat{l}(h^{1:K})$ to reduce the magnitude of the learning signal:

$$\frac{\partial}{\partial\theta}\mathcal{L}^K(x) \simeq \sum_j (\log \hat{l}(h^{1:K}) - b(x)) \frac{\partial}{\partial\theta}\log Q(h^j|x) + \sum_j \tilde{w}^j \frac{\partial}{\partial\theta}\log f(x,h^j),$$

where $h^j \sim Q(h|x)$.

Drawback: the same learning signal is used for all h^j, even though some samples will be much better than others.

VIMCO gradient estimator

Can avoid having to learn an additional predictor by taking advantage of having multiple latent samples for each observation and estimating each $f(x,h^j)$ term from the other $K-1$ terms. We do this using geometric averaging:

$$\hat{f}(x,h^{-j}) = \exp\left(\frac{1}{K-1}\sum_{k\neq j}\log f(x,h^k)\right).$$

This gives us the following learning signal for sample h^j:

$$\hat{L}(h^j|h^{-j}) = \log\frac{1}{K}\sum_{k=1}^{K} f(x,h^k) - \log\frac{1}{K}\left(\sum_{k\neq j} f(x,h^k) + \hat{f}(x,h^{-j})\right),$$

which we use in place of $\log \hat{l}(h^{1:K})$ to obtain the VIMCO gradient estimator:

$$\frac{\partial}{\partial\theta}\mathcal{L}^K(x) \simeq \sum_j \hat{L}(h^j|h^{-j})\frac{\partial}{\partial\theta}\log Q(h^j|x) + \sum_j \tilde{w}^j \frac{\partial}{\partial\theta}\log f(x,h^j).$$

Results: generative modelling

Task: learning a density model of 28×28 binary MNIST digit images.
Model: sigmoid belief network with 3 hidden layers of 200 binary units.
Training algorithms: VIMCO, NVIL, and Reweighted Wake Sleep.

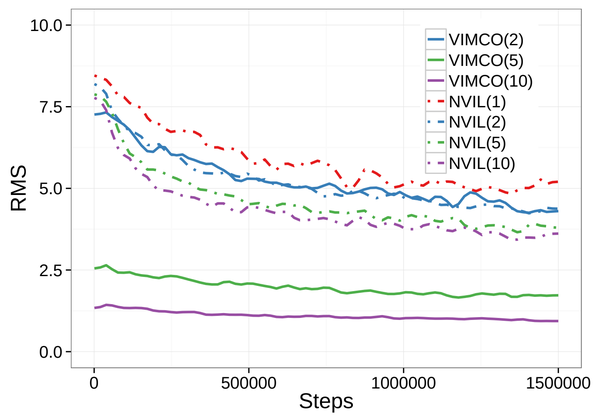

Figure: Learning signal RMS

Number of samples	Training alg. VIMCO	NVIL	RWS
1	—	95.2	—
2	93.5	93.6	94.6
5	92.8	93.7	93.4
10	92.6	93.4	93.0
50	91.9	96.2	92.5

Table: Test neg. log-likelihood

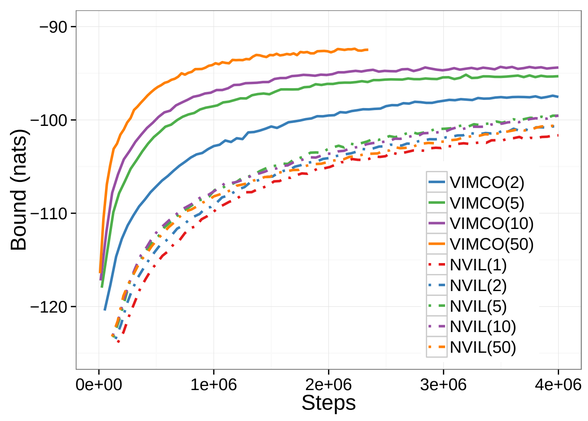

Figure: VIMCO vs. NVIL

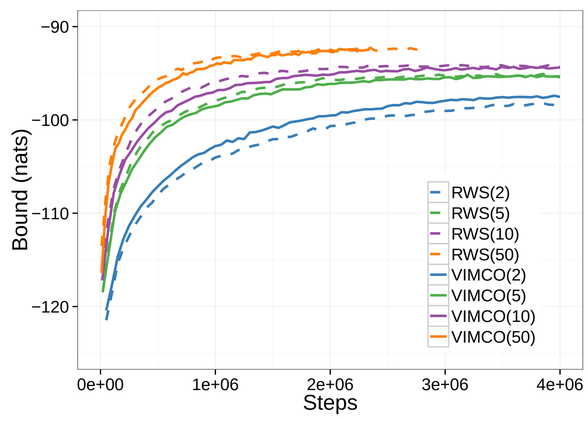

Figure: VIMCO vs. RWS

Results: structured prediction

Task: learning to predict the bottom half of an MNIST digit from its top half.
Model: sigmoid belief network with 2 hidden layers of 200 binary units.
Training algorithms: VIMCO and NVIL.

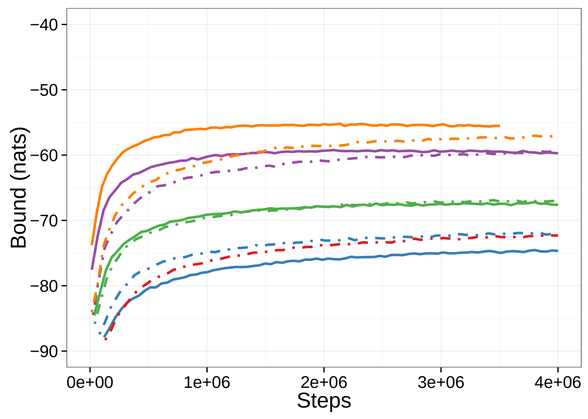

Figure: Prior inference

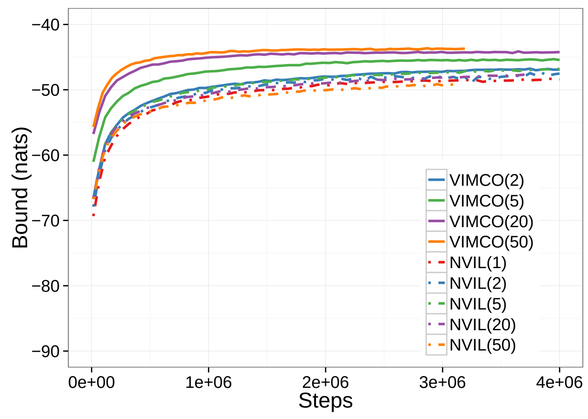

Figure: Posterior inference

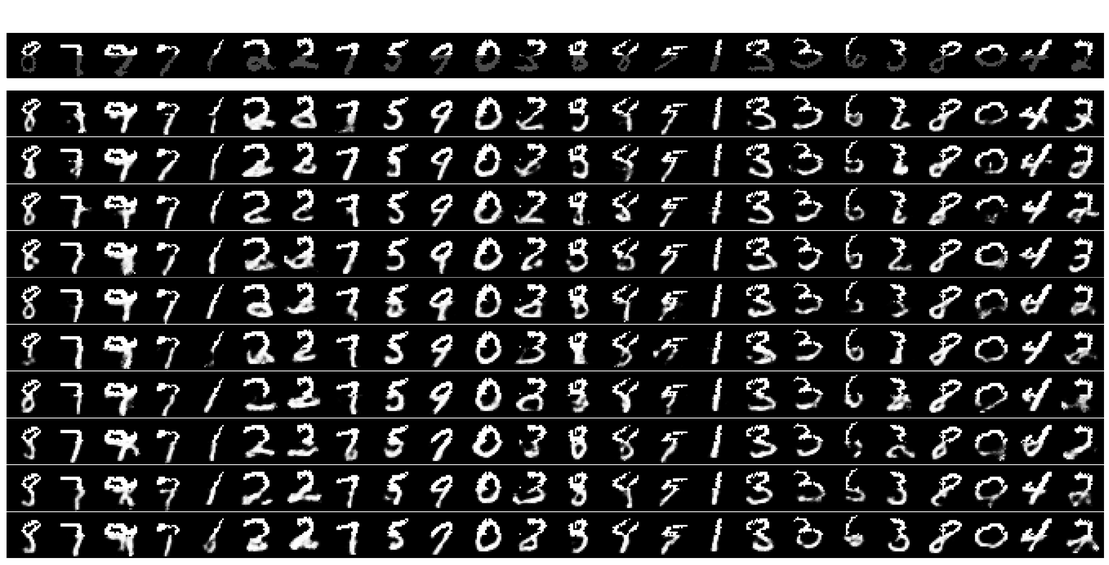

Figure: Conditional completions generated by sampling from an SBN trained using VIMCO with the 20-sample objective.

References

1. Burda, Yuri, Grosse, Roger, and Salakhutdinov, Ruslan. Importance weighted autoencoders. ICLR 2016.
2. Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. ICML 2014.