Neural Variational Inference and Learning

Andriy Mnih, Karol Gregor

22 June 2014
Introduction

- Training directed latent variable models is difficult because inference in them is intractable.
  - Both MCMC and traditional variational methods involve iterative procedures for each datapoint.
- A promising new way to train directed latent variable models:
  - Use feedforward approximation to inference to implement efficient sampling from the variational posterior.
- We propose a general version of this approach that
  1. Can handle both discrete and continuous latent variables.
  2. Does not require any model-specific derivations beyond computing gradients w.r.t. parameters.
High-level overview

- A general approach to variational inference based on three ideas:
  1. Approximating the posterior using highly expressive feed-forward inference networks (e.g. neural nets).
     - These have to be efficient to evaluate and sample from.
  2. Using gradient-based updates to improve the variational bound.
  3. Computing the gradients using samples from the inference net.

- Key: The inference net implements efficient sampling from the approximate posterior.
Given a directed latent variable model that naturally factorizes as

\[ P_\theta(x, h) = P_\theta(x|h)P_\theta(h), \]

We can lower-bound the contribution of \( x \) to the log-likelihood as follows:

\[
\log P_\theta(x) \geq E_Q [\log P_\theta(x, h) - \log Q_\phi(h|x)] = \mathcal{L}_{\theta,\phi}(x),
\]

where \( Q_\phi(h|x) \) is an arbitrary distribution.

In the context of variational inference, \( Q_\phi(h|x) \) is called the variational posterior.
Variational inference (II)

Variational learning involves alternating between maximizing the lower bound $\mathcal{L}_{\theta,\phi}(x)$ w.r.t. the variational distribution $Q_{\phi}(h|x)$ and model parameters $\theta$.

Typically variational inference requires:

- Variational distributions $Q$ with simple factored form and no parameter sharing between distributions for different $x$.
- Simple models $P_{\theta}(x, h)$ yielding tractable expectations.
- Iterative optimization to compute $Q$ for each $x$.

We would like to avoid iterative inference, while allowing expressive, potentially multimodal, posteriors, and highly expressive models.
Neural variational inference and learning (NVIL)

- We achieve these goals by using a feed-forward model for $Q_\phi(h|x)$, making the dependence of the approximate posterior on the input $x$ parametric.
  - This allows us to sample from $Q_\phi(h|x)$ very efficiently.
  - We will refer to $Q$ as the **inference network** because it implements approximate inference for the model being trained.

- We train the model by (locally) maximizing the variational bound $\mathcal{L}_{\theta,\phi}(x)$ w.r.t. $\theta$ and $\phi$.
  - We compute all the required expectations using samples from $Q$. 

The gradients of the bound w.r.t. to the model and inference net parameters are:

\[
\frac{\partial}{\partial \theta} L_{\theta, \phi}(x) = E_Q \left[ \frac{\partial}{\partial \theta} \log P_{\theta}(x, h) \right],
\]

\[
\frac{\partial}{\partial \phi} L_{\theta, \phi}(x) = E_Q \left[ (\log P_{\theta}(x, h) - \log Q_{\phi}(h|x)) \frac{\partial}{\partial \phi} \log Q_{\phi}(h|x) \right].
\]

Note that the learning signal for the inference net is

\[ I_{\phi}(x, h) = \log P_{\theta}(x, h) - \log Q_{\phi}(h|x). \]

This signal is effectively the same as \( \log P_{\theta}(h|x) - \log Q_{\phi}(h|x) \) (up to a constant w.r.t. \( h \)), but is tractable to compute.

The price to pay for tractability is the high variance of the resulting estimates.
Parameter updates

- Given an observation $x$, we can estimate the gradients using Monte Carlo:
  
  1. Sample $h \sim Q_\phi(h|x)$
  2. Compute

     \[
     \frac{\partial}{\partial \theta} \mathcal{L}_{\theta,\phi}(x) \approx \frac{\partial}{\partial \theta} \log P_\theta(x, h)
     \]

     \[
     \frac{\partial}{\partial \phi} \mathcal{L}_{\theta,\phi}(x) \approx (\log P_\theta(x, h) - \log Q_\phi(h|x)) \frac{\partial}{\partial \phi} \log Q_\phi(h|x)
     \]

- Problem: The resulting estimator of the inference network gradient is too high-variance to be useful in practice.

- It can be made practical, however, using several simple model-independent variance reduction techniques.
Reducing variance (I)

- **Key observation**: if $h$ is sampled from $Q_\phi(h|x)$,

$$
(\log P_\theta(x, h) - \log Q_\phi(h|x) - b) \frac{\partial}{\partial \phi} \log Q_\phi(h|x)
$$

is an unbiased estimator of $\frac{\partial}{\partial \phi} \mathcal{L}_{\theta,\phi}(x)$ for any $b$ independent of $h$.

- However, the variance of the estimator does depend on $b$, which allows us to obtain lower-variance estimators by choosing $b$ carefully.

- Our strategy is to choose $b$ so that the resulting learning signal $\log P_\theta(x, h) - \log Q_\phi(h|x) - b$ is close to zero.

- Borrowing terminology from reinforcement learning, we call $b$ a **baseline**.
Reducing variance (II)

Techniques for reducing estimator variance:

1. **Constant baseline**: \( b = \text{a running estimate of the mean of} \)
   \[ l_\phi(x, h) = \log P_\theta(x, h) - \log Q_\phi(h|x). \]
   - Makes the learning signal zero-mean.
   - Enough to obtain reasonable models on MNIST.

2. **Input-dependent baseline**: \( b_\psi(x) \).
   - Can be seen as capturing \( \log P_\theta(x) \).
   - An MLP with a single real-valued output.
   - Makes learning considerably faster and leads to better results.

3. **Variance normalization**: scale the learning signal to unit variance.
   - Can be seen as simple global learning rate adaptation.
   - Makes learning faster and more robust.

4. **Local learning signals**:
   - Take advantage of the Markov properties of the models.
Effects of variance reduction

Sigmoid belief network with two hidden layers of 200 units on MNIST.
Document modelling results

- Task: model the joint distribution of word counts in bags of words describing documents.
- Models: SBN and fDARN models with one hidden layer
- Datasets:
  - 20 Newsgroups: 11K documents, 2K vocabulary
  - Reuters RCV1: 800K documents, 10K vocabulary
- Performance metric: perplexity

<table>
<thead>
<tr>
<th>MODEL</th>
<th>DIM</th>
<th>20 NEWS</th>
<th>REUTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBN</td>
<td>50</td>
<td>909</td>
<td>784</td>
</tr>
<tr>
<td>fDARN</td>
<td>50</td>
<td>917</td>
<td>724</td>
</tr>
<tr>
<td>fDARN</td>
<td>200</td>
<td></td>
<td>598</td>
</tr>
<tr>
<td>LDA</td>
<td>50</td>
<td>1091</td>
<td>1437</td>
</tr>
<tr>
<td>LDA</td>
<td>200</td>
<td>1058</td>
<td>1142</td>
</tr>
<tr>
<td>RepSoftMax</td>
<td>50</td>
<td>953</td>
<td>988</td>
</tr>
<tr>
<td>DocNade</td>
<td>50</td>
<td>896</td>
<td>742</td>
</tr>
</tbody>
</table>
Conclusions

- NVIL is a simple and general training method for directed latent variable models.
  - Can handle both continuous and discrete latent variables.
  - Easy to apply, requiring no model-specific derivations beyond gradient computation.
- Promising document modelling results with DARN and SBN models.
Thank you!