Neural Variational Inference and Learning

Andriy Mnih

Google DeepMind

Deep Probabilistic Models Workshop, Sheffield

2 October 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. Can handle non-linear dependencies in the variational posterior.
    3. 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.

Variational inference (I)

- 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:

$$\begin{aligned} \log P_\theta(x) &\geq E_Q\left[\log P_\theta(x, h) - \log Q_\phi(h|x)\right] \\ &= \mathcal{L}_{\theta,\phi}(x), \end{aligned}$$

  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 θ.
- 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. θ and ϕ.
  - We compute all the required expectations using samples from Q.

Relationship to the existing methods

- Unlike traditional applications of variational inference, NVIL
  - uses a feed-forward network to compute the posterior efficiently,
  - uses sampling to estimate the expectations needed for learning,
  - performs gradient-based updates instead of closed-form ones.
- Unlike the wake-sleep algorithm, NVIL
  - does not require sampling from the model and
  - optimizes a well-defined objective function (the variational bound).

Gradients of the variational bound

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

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

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

- Note that the learning signal for the inference net is $l_\phi(x,h) = \log P_\theta(x,h) - \log Q_\phi(h|x)$.
- 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

$$\begin{aligned}\frac{\partial}{\partial\theta}\mathcal{L}_{\theta,\phi}(x) &\approx \frac{\partial}{\partial\theta}\log P_\theta(x,h)\\ \frac{\partial}{\partial\phi}\mathcal{L}_{\theta,\phi}(x) &\approx (\log P_\theta(x,h) - \log Q_\phi(h|x))\,\frac{\partial}{\partial\phi}\log Q_\phi(h|x)\end{aligned}$$

- 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 $Q_\phi(h|x)$,

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

  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:

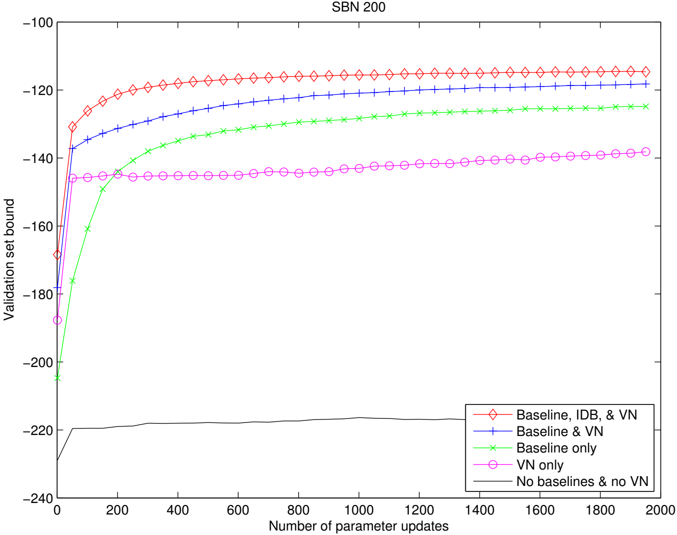
1. Constant baseline: b = a running estimate of the mean of $l_\phi(x, h) = \log P_\theta(x, h) - \log Q_\phi(h|x)$.
   - Makes the learning signal zero-mean.
   - Enough to obtain reasonable models on MNIST.
2. Input-dependent baseline: $b_\psi(x)$.
   - Can be seen as capturing $\log P_\theta(x)$.
   - 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 have unit variance.
   - Can be seen as simple global learning rate adaptation.
   - Makes learning more robust and slightly faster.
4. Local learning signals.

Experiments

- Dataset: MNIST
- Models: sigmoid belief networks
- Inference networks: sigmoid belief networks

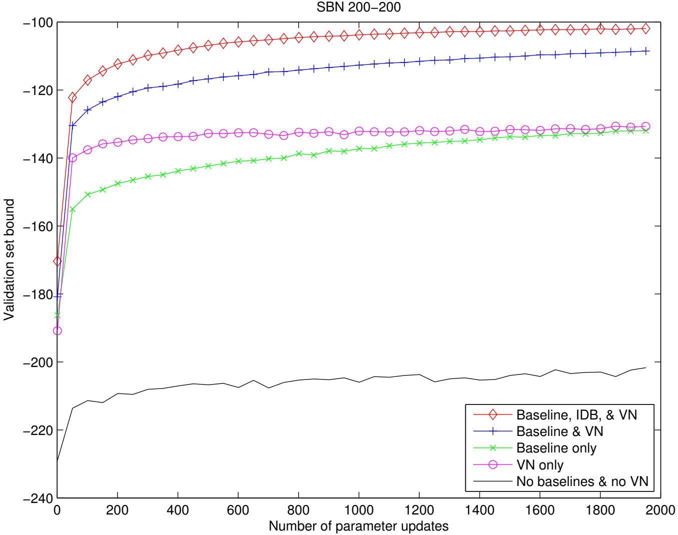
Effects of variance reduction (I)

Sigmoid belief network with one hidden layer of 200 units on MNIST.



Effects of variance reduction (II)

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



NVIL vs. Wake-sleep

NVIL vs. wake-sleep on binarized MNIST:

MODEL	DIM	TEST NLL	
		NVIL	WS
SBN	200	113.1	120.8
SBN	500	112.8	121.4
SBN	200-200	99.8	107.7
SBN	200-200-200	96.7	102.2
SBN	200-200-500	97.0	102.3
fDARN	200	92.5	95.9
fDARN	500	90.7	97.2

- Dim is the number of latent variables in each layer, starting with the deepest one.
- NVIL/WS is the training algorithm.
- NLL is the negative log-likelihood for the tractable models and an estimate of it for the intractable ones.

Non-factorial inference networks

Autoregressive vs. factorial inference networks:

Model	Dim	Test NLL	
		Autoreg	Factorial
SBN	200	103.8	113.1
SBN	500	104.4	112.8
SBN	200-200-200	94.5	96.7
SBN	200-200-500	96.0	97.0

Results on binarized MNIST

Model	Dim	Test NLL
SBN	200	113.1
SBN	500	112.8
SBN	200-200	99.8
SBN	200-200-200	96.7
SBN	200-200-500	97.0
fDARN	200	92.5
fDARN	500	90.7
MoB	500	137.6
RBM (CD3)	500	105.5
RBM (CD25)	500	86.3
NADE	500	88.9

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

MODEL	DIM	20 NEWS	REUTERS
SBN	50	909	784
fDARN	50	917	724
fDARN	200		598
LDA	50	1091	1437
LDA	200	1058	1142
REPSOFTMAX	50	953	988
DOCNADE	50	896	742

Conclusions

- NVIL is a simple and general training method for directed latent variable models.
    - Can handle both continuous and discrete latent variables.
    - Can handle non-linear dependencies in the variational posterior.
    - Easy to apply, requiring no model-specific derivations beyond gradient computation.
- Promising document modelling results with DARN and SBN models.
- For details, see *Neural Variational Inference and Learning in Belief Networks*, Andriy Mnih & Karol Gregor, ICML 2014.

Thank you!