# Neural Variational Inference and Learning in Belief Networks

Andriy Mnih & Karol Gregor
Google DeepMind

## Overview

- We introduce a **simple, efficient, and general method for training directed latent variable models**.
  - Can handle both discrete and continuous latent variables.
  - Easy to apply – requires no model-specific derivations.
- Key idea: Train an auxiliary neural network to perform inference in the model of interest by optimizing the variational bound.
  - Was considered before for Helmholtz machines and rejected as infeasible due to high variance of inference net gradient estimates.
- We make the approach practical using simple and general variance reduction techniques.
- Promising document modelling results using sigmoid belief networks.

## Variational inference

- 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
$$\log P_\theta(x) \geq E_Q\left[\log P_\theta(x,h) - \log Q_\phi(h|x)\right] = \mathcal{L}_{\theta,\phi}(x),$$
where $Q_\phi(h|x)$ is an arbitrary distribution.
- Variational learning involves alternating between maximizing the lower bound $\mathcal{L}_{\theta,\phi}(x)$ w.r.t. the variational distribution/posterior $Q_\phi(h|x)$ and model parameters $\theta$.
- Typically variational inference requires:
  - Variational distributions $Q$ with simple factored form and different parameters for each $x$.
  - Simple models $P_\theta(x,h)$, yielding tractable expectations.
  - Iterative optimization to compute $Q$ for each $x$.

## Neural variational inference and learning (NVIL)

- We propose an approach that avoids iterative inference, while allowing expressive, potentially multimodal, posteriors and highly expressive models.
- This is achieved 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 refer to $Q$ as the *inference network* because it implements approximate inference for the model being trained.
- We train the model and the inference network jointly by updating their parameters to increase the variational lower bound $\mathcal{L}_{\theta,\phi}(x)$.
  - We compute all the required expectations using samples from $Q$.

## Gradients of the variational bound

- The gradients w.r.t. to the model and inference net parameters are:
$$\frac{\partial}{\partial\theta}\mathcal{L}_{\theta,\phi}(x) = E_Q\left[\frac{\partial}{\partial\theta}\log P_\theta(x,h)\right],$$
$$\frac{\partial}{\partial\phi}\mathcal{L}_{\theta,\phi}(x) = E_Q\left[(\log P_\theta(x,h) - \log Q_\phi(h|x))\frac{\partial}{\partial\phi}\log Q_\phi(h|x)\right].$$
- Both gradients can be estimated using samples from the inference net.
- However, the most natural estimator of the inference net gradient is too high-variance to be useful.

## Reducing gradient variance

- 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 $b$ that does not depend on $h$.
- Since the variance of the estimator does depend on $b$, we can obtain estimators with lower variance 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*.

## Variance reduction techniques

1. **Constant baseline** $b$
   - Make $b$ a running estimate of the mean of $\log P_\theta(x,h) - \log Q_\phi(h|x)$.
   - Centers the learning signal, making it approximately zero-mean.
   - Enough to obtain reasonable models on MNIST.
2. **Input-dependent baseline** $b_\psi(x)$
   - An MLP with a single real-valued output.
   - Can be seen as capturing $\log P_\theta(x)$.
   - 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 faster and more robust.
4. **Local learning signals**
   - Simpler, less noisy local learning signals can be derived by taking advantage of the Markov properties of the model and the inference net.
   - Likely to be important for training deeper models.

## Generative modelling of binarized MNIST

### Effect of gradient variance reduction

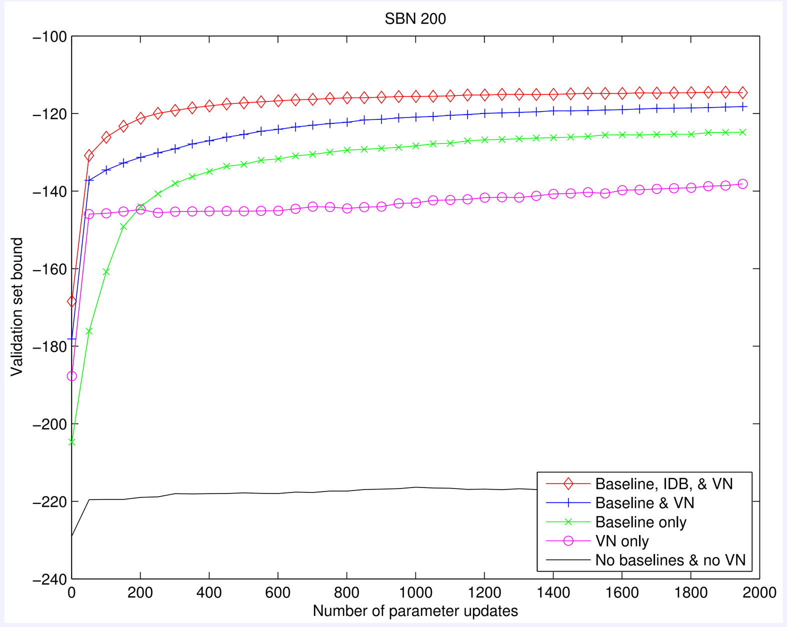


Figure 1: Sigmoid belief network with 1 hidden layer of 200 units.

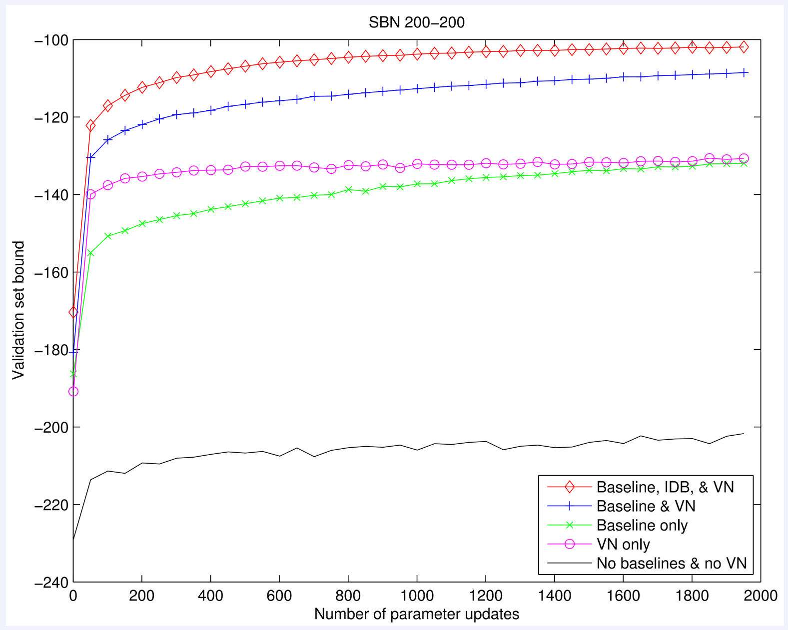


Figure 2: Sigmoid belief network with 2 hidden layers of 200 units.

### NVIL vs. Wake-Sleep

- SBN is a sigmoid belief network.
- fDARN is an SBN with hidden autoregressive connections.
- Dim is the number of latent variables in each layer, starting with the deepest one.
- NVIL and WS refer to NVIL and wake-sleep training respectively.
- NLL is the negative log-likelihood for the tractable models and an estimate of or a bound on it for the intractable ones.

| 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 |
| fDARN | 200 | 92.5 | 95.9 |
| fDARN | 500 | 90.7 | 97.2 |
| fDARN | 400 | 96.3 | |
| DARN | 400 | 93.0 | |
| NADE | 500 | 88.9 | |
| RBM (CD3) | 500 | 105.5 | |
| RBM (CD25) | 500 | 86.3 | |
| MoB | 500 | 137.6 | |

## 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 docs, 2K vocabulary
  - Reuters RCV1
    - 800K docs, 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 |