Reweighted Wake-Sleep

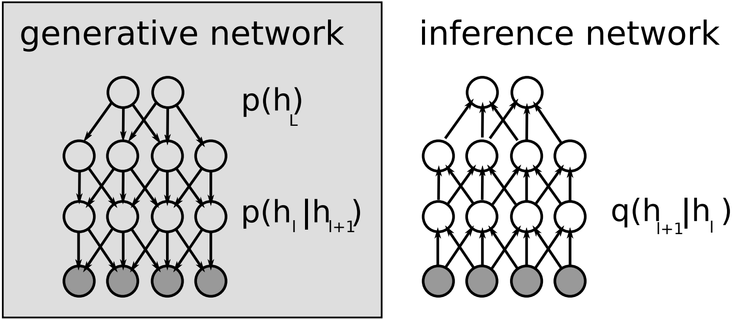
Jörg Bornschein & Yoshua Bengio

University of Montreal & CIFAR

2015/05/08

Helmholtz Machines

We want to train a directed generative model p



$$p(\mathbf{x}, \mathbf{h}) = p(\mathbf{x}|\mathbf{h}_1)p(\mathbf{h}_1|\mathbf{h}_2)...p(\mathbf{h}_L)$$
$$q(\mathbf{h}|\mathbf{x}) = q(\mathbf{h}_1|\mathbf{x})q(\mathbf{h}_2|\mathbf{h}_1)...q(\mathbf{h}_L|\mathbf{h}_{L-1})$$

Helmholtz Machines

Approaches:

- Wake-Sleep algorithm (Frey, Hinton, Dayan, Neal; 1995)
- Neural Variational Inference and Learning (Mnih & Gregor; 2014)
- Variational Autoencoder (Kingma & Welling; 2014 / Renzede et.al.)
- ... *many more* ...

...are typically based on the variational bound of the log-likelihood:

$$\log p(x) \geq \sum_h q(\mathbf{h}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{h}) + H(q(\mathbf{h}|\mathbf{x}))$$

Helmholtz Machines

Approaches:

- Wake-Sleep algorithm (Frey, Hinton, Dayan, Neal; 1995)
- Neural Variational Inference and Learning (Mnih & Gregor; 2014)
- Variational Autoencoder (Kingma & Welling; 2014 / Renzede et.al.)
- ... *many more* ...

...are typically based on the variational bound of the log-likelihood:

$$\log p(x) \geq \sum_h q(\mathbf{h}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{h}) + H(q(\mathbf{h}|\mathbf{x}))$$

Log-Likelihood Estimation

We start with an important sampling estimate:

$$p(\mathbf{x}) = \sum_{\mathbf{h}} q(\mathbf{h} \,|\, \mathbf{x}) \frac{p(\mathbf{x}, \mathbf{h})}{q(\mathbf{h} \,|\, \mathbf{x})} = \mathop{\mathbb{E}}_{\mathbf{h} \sim q(\mathbf{h} \,|\, \mathbf{x})} \left[\frac{p(\mathbf{x}, \mathbf{h})}{q(\mathbf{h} \,|\, \mathbf{x})} \right]$$

$$\simeq \frac{1}{K} \sum_{\substack{k=1 \\ \mathbf{h}^{(k)} \sim q(\mathbf{h} \,|\, \mathbf{x})}}^{K} \frac{p(\mathbf{x}, \mathbf{h}^{(k)})}{q(\mathbf{h}^{(k)} \,|\, \mathbf{x})}$$

- unbiased estimator
- variance depends on the proposal distribution $q(\mathbf{h}|\mathbf{x})$
- minimum variance is obtained with $q(\mathbf{h}|\mathbf{x}) = p(\mathbf{h}|\mathbf{x})$
  (zero variance)

Reweighted Wake-Sleep

From this we can derive an estimator for the parameter gradient:

$$\frac{\partial}{\partial\theta}\mathcal{L}_p \simeq \sum_{k=1}^{K} \tilde{\omega}_k \frac{\partial}{\partial\theta} \log p(\mathbf{x}, \mathbf{h}^{(k)})$$

with $\mathbf{h}^{(k)} \sim q\left(\mathbf{h} \,|\, \mathbf{x}\right)$

and $\tilde{\omega}_k = \frac{\omega_k}{\sum_{k'=1}^{K} \omega_{k'}}$; $\omega_k = \frac{p(\mathbf{x}, \mathbf{h}^{(k)})}{q\left(\mathbf{h}^{(k)} \,|\, \mathbf{x}\right)}$

$=>$ No variational approximation!

Reweighted Wake-Sleep

What is the objective for $q(\mathbf{h}|\mathbf{x})$?

- **Minimize variance!**
- train $q(\mathbf{h}|\mathbf{x})$ to approximate $p(\mathbf{h}|\mathbf{x})$!

What $\mathbf{x}$ do we use when training $q(\mathbf{h}|\mathbf{x})$?

- from the training data set $\mathbf{x} \sim \mathcal{D}$ *(wake phase update)*

  or
- from the current model $\mathbf{x}, \mathbf{h} \sim p(\mathbf{x}, \mathbf{h})$ *(sleep phase update)*

Reweighted Wake-Sleep

Sleep phase q-update:

- consider $\mathbf{x}, \mathbf{h} \sim p(\mathbf{x}, \mathbf{h})$ a fully observed sample
- calculate the gradient $\frac{\partial}{\partial \phi} \mathcal{L}_q = \frac{\partial}{\partial \phi} \log q(\mathbf{h}|\mathbf{x})$

Wake phase q-update:

$$\frac{\partial}{\partial \phi} \mathcal{L}_q \simeq \sum_{k=1}^{K} \tilde{\omega}_k \frac{\partial}{\partial \phi} \log q(\mathbf{x}, \mathbf{h}^{(k)})$$

With the same weights $\tilde{\omega}$ used during the p-update!

Reweighted Wake-Sleep

Sleep phase q-update:

- consider $\mathbf{x}, \mathbf{h} \sim p(\mathbf{x}, \mathbf{h})$ a fully observed sample
- calculate the gradient $\frac{\partial}{\partial \phi} \mathcal{L}_q = \frac{\partial}{\partial \phi} \log q(\mathbf{h}|\mathbf{x})$

Wake phase q-update:

$$\frac{\partial}{\partial \phi} \mathcal{L}_q \simeq \sum_{k=1}^{K} \tilde{\omega}_k \frac{\partial}{\partial \phi} \log q(\mathbf{x}, \mathbf{h}^{(k)})$$

With the same weights $\tilde{\omega}$ used during the p-update!

Reweighted Wake-Sleep

(Reweighted) Wake-Sleep q-update:

$$\arg\min_{\Phi} KL(p_{\Theta}(\mathbf{h}|\mathbf{x}) \mid q_{\Phi}(\mathbf{h}|\mathbf{x}))$$

Variational approaches :

$$\arg\min_{\Phi} KL(q_{\Phi}(\mathbf{h}|\mathbf{x}) \mid p_{\Theta}(\mathbf{h}|\mathbf{x}))$$

RWS with K=1 sample and sleep phase update only is equivalent to classical WS

Reweighted Wake-Sleep

(Reweighted) Wake-Sleep q-update:

$$\arg\min_{\Phi} KL(p_{\Theta}(\mathbf{h}|\mathbf{x}) \mid q_{\Phi}(\mathbf{h}|\mathbf{x}))$$

Variational approaches :

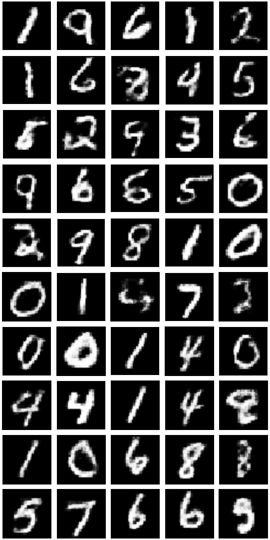
$$\arg\min_{\Phi} KL(q_{\Phi}(\mathbf{h}|\mathbf{x}) \mid p_{\Theta}(\mathbf{h}|\mathbf{x}))$$

RWS with K=1 sample and sleep phase update only is equivalent to classical WS

Empirical results

- using 5-10 samples during training gives significant better results than classical WS
- combining *wake* and *sleep phase* q-updates consistently gives best results
- applied to binarized MNIST, RWS with 5-6 layers results in competitive models

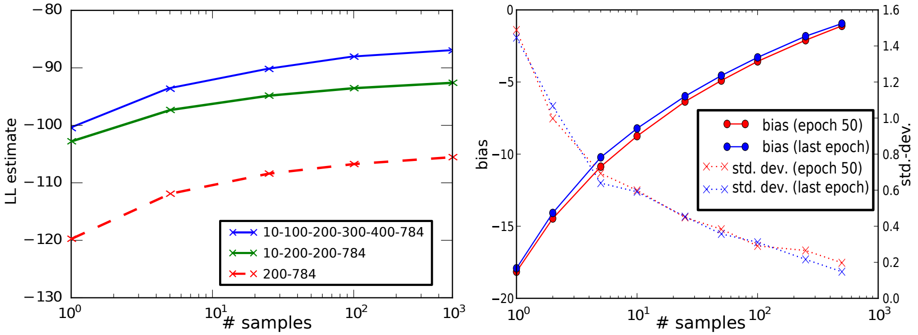
  e.g: 5 hidden layers with 10, 100, 200, 300, 400, 784 units $\Rightarrow$ NLL $\approx$ 85.5



SBN/SBN 10-100-200-300-400

Empirical Results

Sensitivity to number of test samples



Empirical Results

Sensitivity to number of samples during training