Representation Learning — Module 1

Christopher Zach

Department of Electrical Engineering
Chalmers University

Lecture, Sep 2020

Outline

Latent variable models

Latent variable model (LVM)

A LVM is a joint probability distribution $p_\theta(x, z)$, where x is observed in training data but z is not. We only can match the data distribution $p_d(x)$ (via samples) with the marginal of the model

$$p_\theta(x) = \int p_\theta(x, z)\, dz$$

- In most interesting cases computing the marginal $p_\theta(x)$ is intractable

Discussion

When do we expect exact computation of $p_\theta(x)$ to be intractable?

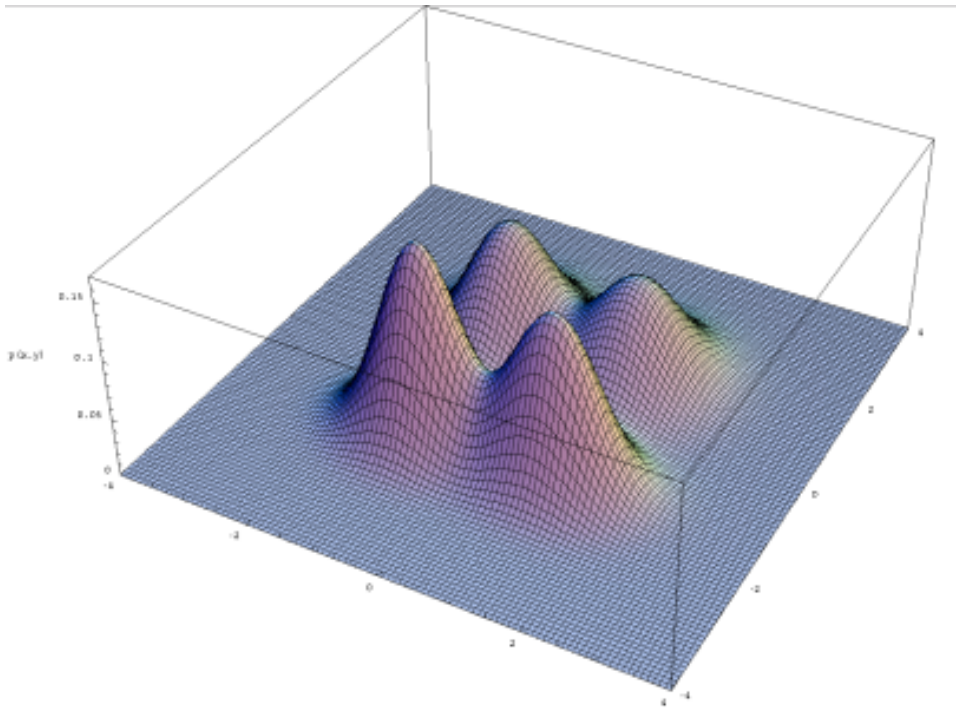
Latent variable models

Examples

- Mixture models

$$p_\theta(x|z) = p_\theta^z(x) \qquad z \in \{1, \ldots, K\} \qquad p(z = k) = w_k$$
$$p_\theta(x) = \sum_k w_k p_\theta^k(x)$$

We cannot observe which mixture component x came from



Latent variable models

Examples (all usually with intractable partition function)

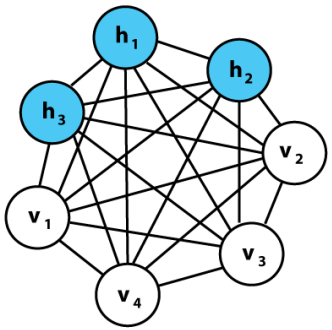
- Boltzmann machines (binary random vectors for visibles x and latent variables z)

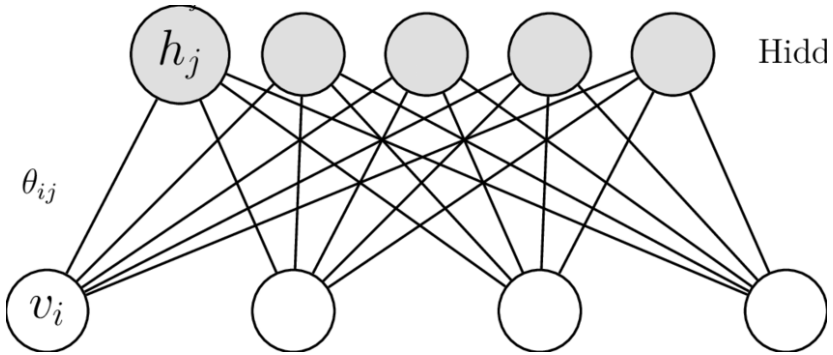
$$p_\theta(x,z) = \frac{1}{Z}\exp\left(\begin{pmatrix}x\\z\end{pmatrix}^T W\begin{pmatrix}x\\z\end{pmatrix} + a^T x + b^T z\right) \qquad \operatorname{diag}(W) = 0$$

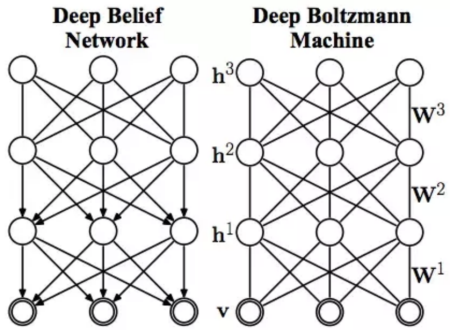
- Restricted Boltzmann machines

$$p_\theta(x,z) = \frac{1}{Z}\exp\left(x^T W z + a^T x + b^T z\right)$$

- Deep belief nets and deep Boltzmann machines







Latent variable models

Examples (all usually with intractable partition function)

- Markov random fields / conditional random fields

$$P(z|x) = \frac{1}{Z} \exp\left(-\sum_s \phi_s(z_s|x_s) - \sum_{s\sim t} \phi_{st}(z_s, z_t)\right)$$

- Topic models
  - Probabilistic latent semantic analysis (PLSA)

$$P(w, d) = P(d) \sum_c P(c|d)P(w|c)$$

  - Latent Dirichlet allocation (LDA)

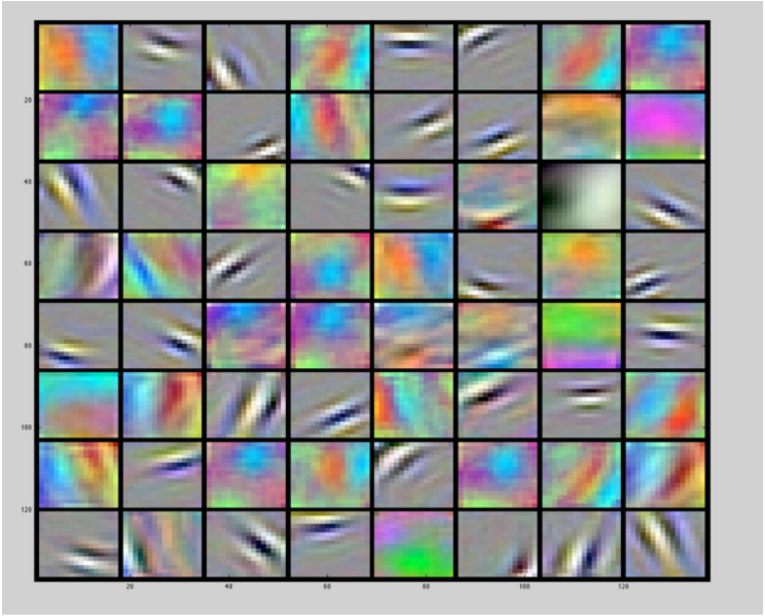
Latent variable models

Sometimes we optimize over latent variables instead of marginalization

- Sparse dictionary learning / sparse coding

$$-\log p_\theta(x) \doteq \min_z \|x - \mathsf{D}z\|^2 + \lambda \|z\|_\varepsilon \qquad \varepsilon \in [0, 1]$$

- Important tool to model images / image patches
- Motivated by the primary visual cortex (V1)
- Usually not interpreted as probabilistic model (but seen as EBM)



Latent variable models

- Recall

$$p_\theta(x) = \int p_\theta(x, z)\, dz$$

- How can we work with latent variables?
- Expectation-maximization algorithm
  - Infer latent variables z for current parameter
  - Optimize log-likelihood with infered z
- Learn jointly a posterior $q(z|x)$
  - Variational Bayes, variational inference
  - Variational auto-encoder

Expectation maximization

- Operational description

$$Q(\theta; \theta^{(t)}) \leftarrow \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta; \theta^{(t)}) \qquad \text{M-step}$$

- E-step: infer latent variables using current estimate of the posterior
- M-step: maximize likelihood using the now fixed posterior
- In one line

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right]$$

- Think of x as complete training set
  - $\log p_\theta(x)$ is sum over training samples

Expectation maximization

- Operational description

$$Q(\theta; \theta^{(t)}) \leftarrow \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta; \theta^{(t)}) \qquad \text{M-step}$$

- E-step: infer latent variables using current estimate of the posterior
- M-step: maximize likelihood using the now fixed posterior
- In one line

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right]$$

- Think of x as complete training set
  - $\log p_\theta(x)$ is sum over training samples

Expectation maximization

- Operational description

$$Q(\theta; \theta^{(t)}) \leftarrow \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta; \theta^{(t)}) \qquad \text{M-step}$$

- E-step: infer latent variables using current estimate of the posterior
- M-step: maximize likelihood using the now fixed posterior
- In one line

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} \mathbb{E}_{z \sim p_{\theta^{(t)}}(z|x)} \left[\log p_\theta(x, z) \right]$$

- Think of x as complete training set
  - $\log p_\theta(x)$ is sum over training samples

Expectation maximization

- Recall

$$Q(\theta;\theta^{(t)}) \leftarrow \mathbb{E}_{z\sim p_{\theta^{(t)}}(z|x)}\left[\log p_\theta(x,z)\right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta;\theta^{(t)}) \qquad \text{M-step}$$

- $Q(\theta;\theta')$ is lower bound of $\log p_\theta(x)$

$$\log p_\theta(x) = \log \frac{p_\theta(x)p_\theta(x,z)}{p_\theta(x,z)} = \log \frac{p_\theta(x,z)}{p_\theta(z|x)} \qquad \forall z : p_\theta(z|x) > 0$$
$$= \log p_\theta(x,z) - \log p_\theta(z|x)$$

Take the expectation w.r.t. $p_{\theta'}(z|x)$

$$\log p_\theta(x) = \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z) - \log p_\theta(z|x)\right]$$
$$= \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z)\right] - \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]$$
$$= Q(\theta;\theta') - \underbrace{\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{\leq 0} \geq Q(\theta;\theta')$$

Expectation maximization

- Recall

$$Q(\theta;\theta^{(t)}) \leftarrow \mathbb{E}_{z\sim p_{\theta^{(t)}}(z|x)}\left[\log p_\theta(x,z)\right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta;\theta^{(t)}) \qquad \text{M-step}$$

- $Q(\theta;\theta')$ is lower bound of $\log p_\theta(x)$

$$\log p_\theta(x) = \log \frac{p_\theta(x)p_\theta(x,z)}{p_\theta(x,z)} = \log \frac{p_\theta(x,z)}{p_\theta(z|x)} \qquad \forall z : p_\theta(z|x) > 0$$
$$= \log p_\theta(x,z) - \log p_\theta(z|x)$$

Take the expectation w.r.t. $p_{\theta'}(z|x)$

$$\log p_\theta(x) = \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z) - \log p_\theta(z|x)\right]$$
$$= \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z)\right] - \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]$$
$$= Q(\theta;\theta') - \underbrace{\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{\leq 0} \geq Q(\theta;\theta')$$

Expectation maximization

- Recall

$$Q(\theta;\theta^{(t)}) \leftarrow \mathbb{E}_{z\sim p_{\theta^{(t)}}(z|x)}\left[\log p_\theta(x,z)\right] \qquad \text{E-step}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta'} Q(\theta;\theta^{(t)}) \qquad \text{M-step}$$

- $Q(\theta;\theta')$ is lower bound of $\log p_\theta(x)$

$$\log p_\theta(x) = \log \frac{p_\theta(x)p_\theta(x,z)}{p_\theta(x,z)} = \log \frac{p_\theta(x,z)}{p_\theta(z|x)} \qquad \forall z : p_\theta(z|x) > 0$$
$$= \log p_\theta(x,z) - \log p_\theta(z|x)$$

Take the expectation w.r.t. $p_{\theta'}(z|x)$

$$\log p_\theta(x) = \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z) - \log p_\theta(z|x)\right]$$
$$= \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(x,z)\right] - \mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]$$
$$= Q(\theta;\theta') - \underbrace{\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{\leq 0} \geq Q(\theta;\theta')$$

Expectation maximization

- Cross entropy between p and q

$$H(p,q) = -\mathbb{E}_{x\sim p}\left[\log q(x)\right] = H(p) + D_{KL}(p\|q) \geq H(p) = H(p,p)$$

- Recall

$$\log p_\theta(x) = Q(\theta;\theta') \underbrace{-\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{=H(p_{\theta'}(z|x),p_\theta(z|x))}$$

- Therefore

$$\log p_\theta(x) = Q(\theta;\theta') + H(p_{\theta'}(z|x), p_\theta(z|x))$$
$$\log p_{\theta'}(x) = Q(\theta';\theta') + H(p_{\theta'}(z|x))$$

- Assume θ maximizes $Q(\theta;\theta')$ for given θ'

$$0 \leq Q(\theta;\theta') - Q(\theta';\theta')$$
$$= \log p_\theta(x) - \log p_{\theta'}(x) + \underbrace{H(p_{\theta'}(z|x)) - H(p_{\theta'}(z|x), p_\theta(z|x))}_{\leq 0} \leq \log p_\theta(x) - \log p_{\theta'}(x)$$

- Updating $\theta^{(t+1)} \leftarrow \arg\max_\theta Q(\theta;\theta^{(t)})$ improves marginal likelihood

$$p_{\theta^{(t+1)}}(x) \geq p_{\theta^{(t)}}(x)$$

Expectation maximization

- Cross entropy between p and q

$$H(p,q) = -\mathbb{E}_{x\sim p}\left[\log q(x)\right] = H(p) + D_{KL}(p\|q) \geq H(p) = H(p,p)$$

- Recall

$$\log p_\theta(x) = Q(\theta;\theta') \underbrace{-\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{=H(p_{\theta'}(z|x),p_\theta(z|x))}$$

- Therefore

$$\log p_\theta(x) = Q(\theta;\theta') + H(p_{\theta'}(z|x), p_\theta(z|x))$$
$$\log p_{\theta'}(x) = Q(\theta';\theta') + H(p_{\theta'}(z|x))$$

- Assume θ maximizes $Q(\theta;\theta')$ for given θ'

$$0 \leq Q(\theta;\theta') - Q(\theta';\theta')$$
$$= \log p_\theta(x) - \log p_{\theta'}(x) + \underbrace{H(p_{\theta'}(z|x)) - H(p_{\theta'}(z|x), p_\theta(z|x))}_{\leq 0} \leq \log p_\theta(x) - \log p_{\theta'}(x)$$

- Updating $\theta^{(t+1)} \leftarrow \arg\max_\theta Q(\theta;\theta^{(t)})$ improves marginal likelihood

$$p_{\theta^{(t+1)}}(x) \geq p_{\theta^{(t)}}(x)$$

Expectation maximization

- Cross entropy between p and q

$$H(p,q) = -\mathbb{E}_{x\sim p}\left[\log q(x)\right] = H(p) + D_{KL}(p\|q) \geq H(p) = H(p,p)$$

- Recall

$$\log p_\theta(x) = Q(\theta;\theta') \underbrace{-\mathbb{E}_{z\sim p_{\theta'}(z|x)}\left[\log p_\theta(z|x)\right]}_{=H(p_{\theta'}(z|x),p_\theta(z|x))}$$

- Therefore

$$\log p_\theta(x) = Q(\theta;\theta') + H(p_{\theta'}(z|x), p_\theta(z|x))$$
$$\log p_{\theta'}(x) = Q(\theta';\theta') + H(p_{\theta'}(z|x))$$

- Assume θ maximizes $Q(\theta;\theta')$ for given θ'

$$0 \leq Q(\theta;\theta') - Q(\theta';\theta')$$
$$= \log p_\theta(x) - \log p_{\theta'}(x) + \underbrace{H(p_{\theta'}(z|x)) - H(p_{\theta'}(z|x), p_\theta(z|x))}_{\leq 0} \leq \log p_\theta(x) - \log p_{\theta'}(x)$$

- Updating $\theta^{(t+1)} \leftarrow \arg\max_\theta Q(\theta;\theta^{(t)})$ improves marginal likelihood

$$p_{\theta^{(t+1)}}(x) \geq p_{\theta^{(t)}}(x)$$

Expectation maximization

EM as alternating optimization

- Consider

$$F(q,\theta) := \log p_\theta(x) - D_{KL}(q||p_\theta(\cdot|x))$$
$$= \log p_\theta(x) - \int q(z) \log \frac{q(z)}{p_\theta(z|x)}\, dz$$
$$= \mathbb{E}_{z\sim q}\left[\log p_\theta(x) - \log \frac{q(z)}{p_\theta(z|x)}\right] = \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x)p_\theta(z|x)}{q(z)}\right]$$
$$= \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x,z)}{q(z)}\right] = \mathbb{E}_{z\sim q}\left[\log p_\theta(x,z)\right] + H(q)$$

- Optimize w.r.t. q and θ by alternating

$$q^{(t+1)} \leftarrow \arg\max_q -D_{KL}(q||p_\theta(\cdot|x)) = p_\theta(\cdot|x)$$
$$\theta^{(t+1)} \leftarrow \arg\max_\theta \mathbb{E}_{z\sim q^{(t+1)}}\left[\log p_\theta(x,z)\right]$$

- This interpretation is useful when exact optimization w.r.t. q (E-step) is hard

Expectation maximization

EM as alternating optimization

- Consider

$$
\begin{aligned}
F(q,\theta) &:= \log p_\theta(x) - D_{KL}(q\|p_\theta(\cdot|x)) \\
&= \log p_\theta(x) - \int q(z) \log \frac{q(z)}{p_\theta(z|x)}\, dz \\
&= \mathbb{E}_{z\sim q}\left[\log p_\theta(x) - \log \frac{q(z)}{p_\theta(z|x)}\right] = \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x)p_\theta(z|x)}{q(z)}\right] \\
&= \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x,z)}{q(z)}\right] = \mathbb{E}_{z\sim q}\left[\log p_\theta(x,z)\right] + H(q)
\end{aligned}
$$

- Optimize w.r.t. q and θ by alternating

$$
\begin{aligned}
q^{(t+1)} &\leftarrow \arg\max_q -D_{KL}(q\|p_\theta(\cdot|x)) = p_\theta(\cdot|x) \\
\theta^{(t+1)} &\leftarrow \arg\max_\theta \mathbb{E}_{z\sim q^{(t+1)}}\left[\log p_\theta(x,z)\right]
\end{aligned}
$$

- This interpretation is useful when exact optimization w.r.t. q (E-step) is hard

Expectation maximization

EM as alternating optimization

- Consider

$$
\begin{aligned}
F(q,\theta) &:= \log p_\theta(x) - D_{KL}(q\|p_\theta(\cdot|x)) \\
&= \log p_\theta(x) - \int q(z) \log \frac{q(z)}{p_\theta(z|x)}\, dz \\
&= \mathbb{E}_{z\sim q}\left[\log p_\theta(x) - \log \frac{q(z)}{p_\theta(z|x)}\right] = \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x)p_\theta(z|x)}{q(z)}\right] \\
&= \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x,z)}{q(z)}\right] = \mathbb{E}_{z\sim q}\left[\log p_\theta(x,z)\right] + H(q)
\end{aligned}
$$

- Optimize w.r.t. q and θ by alternating

$$
\begin{aligned}
q^{(t+1)} &\leftarrow \arg\max_q -D_{KL}(q\|p_\theta(\cdot|x)) = p_\theta(\cdot|x) \\
\theta^{(t+1)} &\leftarrow \arg\max_\theta \mathbb{E}_{z\sim q^{(t+1)}}\left[\log p_\theta(x,z)\right]
\end{aligned}
$$

- This interpretation is useful when exact optimization w.r.t. q (E-step) is hard

Expectation maximization

EM as alternating optimization

- Consider

$$
\begin{aligned}
F(q,\theta) &:= \log p_\theta(x) - D_{KL}(q\|p_\theta(\cdot|x)) \\
&= \log p_\theta(x) - \int q(z) \log \frac{q(z)}{p_\theta(z|x)}\, dz \\
&= \mathbb{E}_{z\sim q}\left[\log p_\theta(x) - \log \frac{q(z)}{p_\theta(z|x)}\right] = \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x)p_\theta(z|x)}{q(z)}\right] \\
&= \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x,z)}{q(z)}\right] = \mathbb{E}_{z\sim q}\left[\log p_\theta(x,z)\right] + H(q)
\end{aligned}
$$

- Optimize w.r.t. q and θ by alternating

$$
\begin{aligned}
q^{(t+1)} &\leftarrow \arg\max_q -D_{KL}(q\|p_\theta(\cdot|x)) = p_\theta(\cdot|x) \\
\theta^{(t+1)} &\leftarrow \arg\max_\theta \mathbb{E}_{z\sim q^{(t+1)}}\left[\log p_\theta(x,z)\right]
\end{aligned}
$$

- This interpretation is useful when exact optimization w.r.t. q (E-step) is hard

Expectation maximization

EM as alternating optimization

- Consider

$$
\begin{aligned}
F(q,\theta) &:= \log p_\theta(x) - D_{KL}(q\|p_\theta(\cdot|x)) \\
&= \log p_\theta(x) - \int q(z) \log \frac{q(z)}{p_\theta(z|x)}\, dz \\
&= \mathbb{E}_{z\sim q}\left[\log p_\theta(x) - \log \frac{q(z)}{p_\theta(z|x)}\right] = \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x)p_\theta(z|x)}{q(z)}\right] \\
&= \mathbb{E}_{z\sim q}\left[\log \frac{p_\theta(x,z)}{q(z)}\right] = \mathbb{E}_{z\sim q}\left[\log p_\theta(x,z)\right] + H(q)
\end{aligned}
$$

- Optimize w.r.t. q and θ by alternating

$$
\begin{aligned}
q^{(t+1)} &\leftarrow \arg\max_q -D_{KL}(q\|p_\theta(\cdot|x)) = p_\theta(\cdot|x) \\
\theta^{(t+1)} &\leftarrow \arg\max_\theta \mathbb{E}_{z\sim q^{(t+1)}}\left[\log p_\theta(x,z)\right]
\end{aligned}
$$

- This interpretation is useful when exact optimization w.r.t. q (E-step) is hard

Evidence lower bound (ELBO)

- One of *the* workhorses in ML
- Closely linked with EM algorithm
- ELBO: evidence lower bound

$$\log p_\theta(x) = \log \int p_\theta(x,z)\,dz \qquad \text{Def. of marginal prob.}$$

$$= \log \int \frac{q(z|x)p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Insert } 1 = \frac{q(z|x)}{q(z|x)}$$

$$\geq \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \qquad \text{Identify expectation}$$

$$= \underbrace{\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x))}_{=:\mathcal{L}} \qquad \text{Identify entropy}$$

Evidence lower bound (ELBO)

- One of *the* workhorses in ML
- Closely linked with EM algorithm
- ELBO: evidence lower bound

$$\log p_\theta(x) = \log \int p_\theta(x,z)\,dz \qquad \text{Def. of marginal prob.}$$

$$= \log \int \frac{q(z|x)p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Insert } 1 = \frac{q(z|x)}{q(z|x)}$$

$$\geq \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \qquad \text{Identify expectation}$$

$$= \underbrace{\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x))}_{=:\mathcal{L}} \qquad \text{Identify entropy}$$

Evidence lower bound (ELBO)

- One of *the* workhorses in ML
- Closely linked with EM algorithm
- ELBO: evidence lower bound

$$\log p_\theta(x) = \log \int p_\theta(x,z)\,dz \qquad \text{Def. of marginal prob.}$$

$$= \log \int \frac{q(z|x)p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Insert } 1 = \frac{q(z|x)}{q(z|x)}$$

$$\geq \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \qquad \text{Identify expectation}$$

$$= \underbrace{\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x))}_{=:\mathcal{L}} \qquad \text{Identify entropy}$$

Evidence lower bound (ELBO)

- One of *the* workhorses in ML
- Closely linked with EM algorithm
- ELBO: evidence lower bound

$$\log p_\theta(x) = \log \int p_\theta(x,z)\,dz \qquad \text{Def. of marginal prob.}$$

$$= \log \int \frac{q(z|x)p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Insert } 1 = \frac{q(z|x)}{q(z|x)}$$

$$\geq \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)}\,dx \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \qquad \text{Identify expectation}$$

$$= \underbrace{\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x))}_{=:\mathcal{L}} \qquad \text{Identify entropy}$$

Evidence lower bound (ELBO)

- One of *the* workhorses in ML
- Closely linked with EM algorithm
- ELBO: evidence lower bound

$$
\begin{aligned}
\log p_\theta(x) &= \log \int p_\theta(x,z)\,dz && \text{Def. of marginal prob.} \\
&= \log \int \frac{q(z|x)p_\theta(x,z)}{q(z|x)}\,dx && \text{Insert } 1 = \frac{q(z|x)}{q(z|x)} \\
&\geq \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)}\,dx && \text{Jensen ineq.} \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] && \text{Identify expectation} \\
&= \underbrace{\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x))}_{=:\mathcal{L}} && \text{Identify entropy}
\end{aligned}
$$

Evidence lower bound (ELBO)

- ELBO $\mathcal{L}$ can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x)) \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}\right] \\
&= \log p_\theta(x) + \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)}{q(z|x)}\right] \\
&= \log p_\theta(x) - \underbrace{D_{KL}(q(\cdot|x)\|p_\theta(\cdot|x))}_{\geq 0} \leq \log p_\theta(x)
\end{aligned}$$

- $\mathcal{L} = \log p_\theta(x)$ (ELBO is tight) iff $q(\cdot|x) = p_\theta(\cdot|x)$ a.e.

Evidence lower bound (ELBO)

- ELBO $\mathcal{L}$ can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x)) \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}\right] \\
&= \log p_\theta(x) + \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)}{q(z|x)}\right] \\
&= \log p_\theta(x) - \underbrace{D_{KL}(q(\cdot|x)\|p_\theta(\cdot|x))}_{\geq 0} \leq \log p_\theta(x)
\end{aligned}$$

- $\mathcal{L} = \log p_\theta(x)$ (ELBO is tight) iff $q(\cdot|x) = p_\theta(\cdot|x)$ a.e.

Evidence lower bound (ELBO)

- ELBO $\mathcal{L}$ can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x)) \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}\right] \\
&= \log p_\theta(x) + \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)}{q(z|x)}\right] \\
&= \log p_\theta(x) - \underbrace{D_{KL}(q(\cdot|x)\|p_\theta(\cdot|x))}_{\geq 0} \leq \log p_\theta(x)
\end{aligned}$$

- $\mathcal{L} = \log p_\theta(x)$ (ELBO is tight) iff $q(\cdot|x) = p_\theta(\cdot|x)$ a.e.

Evidence lower bound (ELBO)

- ELBO $\mathcal{L}$ can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x)) \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}\right] \\
&= \log p_\theta(x) + \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)}{q(z|x)}\right] \\
&= \log p_\theta(x) - \underbrace{D_{KL}(q(\cdot|x)||p_\theta(\cdot|x))}_{\geq 0} \leq \log p_\theta(x)
\end{aligned}$$

- $\mathcal{L} = \log p_\theta(x)$ (ELBO is tight) iff $q(\cdot|x) = p_\theta(\cdot|x)$ a.e.

Evidence lower bound (ELBO)

- ELBO $\mathcal{L}$ can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q(\cdot|x)) \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q(z|x)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}\right] \\
&= \log p_\theta(x) + \mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p_\theta(z|x)}{q(z|x)}\right] \\
&= \log p_\theta(x) - \underbrace{D_{KL}(q(\cdot|x)\|p_\theta(\cdot|x))}_{\geq 0} \leq \log p_\theta(x)
\end{aligned}$$

- $\mathcal{L} = \log p_\theta(x)$ (ELBO is tight) iff $q(\cdot|x) = p_\theta(\cdot|x)$ a.e.

EM and ELBO / variational Bayes

Are EM and VB the same?

- ELBO/VB is more general than EM (IMHO)
- We usually call it EM if the E-step is tractable
  - Clustering, mixture of distributions
- We call it variational Bayes e.g. when
  - we use a simpler parametric approximation $q(\cdot|x)$ for the true posterior $p_\theta(\cdot|x)$
  - we are interested in full posteriors over the parameters $p(\theta|x)$

$$\log p(\theta|x) \geq \mathbb{E}_{\theta\sim q(\cdot|x)}\left[\log p(\theta, x)\right] + H(q(\cdot|x))$$

Outline

Variational auto-encoder (VAE)

Overview

The variational auto-encoder (VAE) / auto-encoding variational Bayes

- uses a generative model for $p_\theta(x, z)$
- identifies the ELBO as sum of reconstruction and regularization terms
- extends auto-encoders with stochastic latent variables
- proposes a method to apply back-prop through stochastic network units

- D.P. Kingma & M. Welling, “Auto-Encoding Variational Bayes”
- Rezende, D. J., S. Mohamed, and D. Wierstra, “Stochastic back-propagation and approximate inference in deep generative models”

Variational auto-encoders (VAE)

- Recall the ELBO (and parametrize $q = q_\phi$ via ϕ)

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q_\phi(\cdot|x)) \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \leq \log p_\theta(x)\end{aligned}$$

- Generative model for $p_\theta(x,z)$

$$p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- p_z user-specified prior over latent variables, e.g.

$$p_z(z) = \mathcal{N}(z;\mathbf{0},\mathrm{I})$$

- Typical model for $p_\theta(x|z)$

$$p_\theta(x|z) = \mathcal{N}(x;\mu_\theta(z),\Sigma_\theta(z))$$

μ_θ and Σ_θ are functions defined via NNs (with parameters θ)

Variational auto-encoders (VAE)

- Recall the ELBO (and parametrize $q = q_\phi$ via ϕ)

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x,z)\right] + H(q_\phi(\cdot|x)) \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \leq \log p_\theta(x)\end{aligned}$$

- Generative model for $p_\theta(x,z)$

$$p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- p_z user-specified prior over latent variables, e.g.

$$p_z(z) = \mathcal{N}(z;\mathbf{0},\mathrm{I})$$

- Typical model for $p_\theta(x|z)$

$$p_\theta(x|z) = \mathcal{N}(x;\mu_\theta(z),\Sigma_\theta(z))$$

  μ_θ and Σ_θ are functions defined via NNs (with parameters θ)

Variational auto-encoders (VAE)

- Recall ELBO and the generative model

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \qquad p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- Insert into $\mathcal{L}$

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x|z)p_z(z)}{q_\phi(z|x)}\right] \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_z(z)}\right] \\
&= \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi} \to \max_{\theta,\phi}
\end{aligned}$$

- Encoder $q_\phi(z|x)$
- Decoder $p_\theta(x|z)$
- ELBO is tight if $q_\phi(z|x) = p_\theta(z|x)$

Variational auto-encoders (VAE)

- Recall ELBO and the generative model

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \qquad p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- Insert into $\mathcal{L}$

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x|z)p_z(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_z(z)}\right] \\ &= \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi} \to \max_{\theta,\phi}\end{aligned}$$

- Encoder $q_\phi(z|x)$
- Decoder $p_\theta(x|z)$
- ELBO is tight if $q_\phi(z|x) = p_\theta(z|x)$

Variational auto-encoders (VAE)

- Recall ELBO and the generative model

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \qquad p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- Insert into $\mathcal{L}$

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x|z)p_z(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_z(z)}\right] \\ &= \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi} \to \max_{\theta,\phi}\end{aligned}$$

- Encoder $q_\phi(z|x)$
- Decoder $p_\theta(x|z)$
- ELBO is tight if $q_\phi(z|x) = p_\theta(z|x)$

Variational auto-encoders (VAE)

- Recall ELBO and the generative model

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \qquad p_\theta(x,z) = p_\theta(x|z)p_z(z)$$

- Insert into $\mathcal{L}$

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x|z)p_z(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_z(z)}\right] \\ &= \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi} \to \max_{\theta,\phi}\end{aligned}$$

- Encoder $q_\phi(z|x)$
- Decoder $p_\theta(x|z)$
- ELBO is tight if $q_\phi(z|x) = p_\theta(z|x)$

Variational auto-encoders (VAE)

- Why "reconstruction term"

$$\mathcal{L} = \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi}$$

- With our choice of $\log p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \Sigma_\theta(z))$

$$\log p_\theta(x|z) \doteq -\frac{1}{2}\left(x-\mu_\theta(z)\right)^T \Sigma_\theta(z)^{-1}\left(x-\mu_\theta(z)\right)$$

- Squared Mahalanobis distance between x and decoded $\mu_\theta(z)$
- With $\log p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \sigma^2 \mathrm{I})$

$$\log p_\theta(x|z) \doteq -\frac{\|x-\mu_\theta(z)\|^2}{2\sigma^2}$$

Variational auto-encoders (VAE)

- Why "reconstruction term"

$$\mathcal{L} = \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)}_{\text{Regularization of } q_\phi}$$

- With our choice of $\log p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \Sigma_\theta(z))$

$$\log p_\theta(x|z) \doteq -\frac{1}{2}\left(x - \mu_\theta(z)\right)^T \Sigma_\theta(z)^{-1}\left(x - \mu_\theta(z)\right)$$

- Squared Mahalanobis distance between x and decoded $\mu_\theta(z)$
- With $\log p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \sigma^2 \mathrm{I})$

$$\log p_\theta(x|z) \doteq -\frac{\|x - \mu_\theta(z)\|^2}{2\sigma^2}$$

Variational auto-encoders (VAE)

- Role of the "regularization term"

$$D_{KL}\left(q_\phi(\cdot|x)||p_z\right)$$

- Favors posterior $q_\phi(\cdot|x)$ to be "disentangled" if $p_z(\cdot) = \mathcal{N}(\cdot; 0, \mathrm{I})$
  - Components of $\mathcal{N}(0, \mathrm{I})$ random vector are independent
- Stronger disentanglement by increasing weight: β-VAE

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - \beta D_{KL}\left(q_\phi(\cdot|x)||p_z\right)$$

  for a $\beta \geq 1$

- Higgins et al., "β-VAE: Learning basic visual concepts with a constrained variational framework"

Variational auto-encoders (VAE)

- How to train a VAE?
- Given training data $\{x_i\}$

$$J(\theta, \phi) = \frac{1}{N} \sum \left(\mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right] - D_{KL}\left(q_\phi(\cdot|x_i) \| p_z \right) \right)$$

- 2nd term often has closed-form expression
  - p_z and $q(\cdot|x_i)$ are Gaussians

$$D_{KL}(\mathcal{N}_0 \| \mathcal{N}_1) \doteq \frac{1}{2} \left(\text{trace}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right)$$

  - $p_z(z) = \mathcal{N}(z; \mathbf{0}, \mathrm{I})$

$$D_{KL}(\mathcal{N}(\mu, \Sigma) \| p_z) \doteq \frac{1}{2} \left(\text{trace}(\Sigma) + \|\mu\|^2 - \log |\Sigma| \right)$$

Variational auto-encoders (VAE)

- How to train a VAE?
- Given training data $\{x_i\}$

$$J(\theta, \phi) = \frac{1}{N} \sum \left(\mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right] - D_{KL}\left(q_\phi(\cdot|x_i) \| p_z \right) \right)$$

- 2nd term often has closed-form expression
  - p_z and $q(\cdot|x_i)$ are Gaussians

$$D_{KL}(\mathcal{N}_0 \| \mathcal{N}_1) \doteq \frac{1}{2} \left(\text{trace}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right)$$

  - $p_z(z) = \mathcal{N}(z; \mathbf{0}, \mathrm{I})$

$$D_{KL}(\mathcal{N}(\mu, \Sigma) \| p_z) \doteq \frac{1}{2} \left(\text{trace}(\Sigma) + \|\mu\|^2 - \log |\Sigma| \right)$$

Variational auto-encoders (VAE)

- How to train a VAE?
- Given training data $\{x_i\}$

$$J(\theta, \phi) = \frac{1}{N} \sum \left(\mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right] - D_{KL}\left(q_\phi(\cdot|x_i) || p_z \right) \right)$$

- 1st term

$$\nabla_\theta \frac{1}{N} \sum \mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right] \qquad \nabla_\phi \frac{1}{N} \sum \mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right]$$

- ∇_θ: MC estimator

$$\nabla_\theta \mathbb{E}_{z \sim q_\phi(\cdot|x_i)} \left[\log p_\theta(x_i|z) \right] \approx \frac{1}{T} \nabla_\theta \log p_\theta(x_i|z^{(t)}) \qquad z^{(t)} \sim q_\phi(z|x_i)$$

  Works okay

Variational auto-encoders (VAE)

- How to train a VAE?
- ∇_ϕ: MC estimator

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] = \nabla_\phi \int q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz$$

$$= \int \frac{q_\phi(z|x_i)}{q_\phi(z|x_i)} \nabla_\phi q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz$$

$$= \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\nabla_\phi \log q_\phi(z|x_i)\right]$$

$$\approx \frac{1}{T}\log p_\theta(x_i|z^{(t)})\nabla_\phi \log q_\phi(z|x_i) \qquad z^{(t)} \sim q_\phi(z|x_i)$$

High variance in practice

Variational auto-encoders (VAE)

- How to train a VAE?
- ∇_ϕ: MC estimator

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] &= \nabla_\phi \int q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \int \frac{q_\phi(z|x_i)}{q_\phi(z|x_i)}\nabla_\phi q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\nabla_\phi \log q_\phi(z|x_i)\right] \\
&\approx \frac{1}{T}\log p_\theta(x_i|z^{(t)})\nabla_\phi \log q_\phi(z|x_i) \qquad z^{(t)} \sim q_\phi(z|x_i)
\end{aligned}$$

High variance in practice

Variational auto-encoders (VAE)

- How to train a VAE?
- ∇_ϕ: MC estimator

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] &= \nabla_\phi \int q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \int \frac{q_\phi(z|x_i)}{q_\phi(z|x_i)}\nabla_\phi q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\nabla_\phi \log q_\phi(z|x_i)\right] \\
&\approx \frac{1}{T}\log p_\theta(x_i|z^{(t)})\nabla_\phi \log q_\phi(z|x_i) \qquad z^{(t)} \sim q_\phi(z|x_i)
\end{aligned}$$

High variance in practice

Variational auto-encoders (VAE)

- How to train a VAE?
- ∇_ϕ: MC estimator

$$
\begin{aligned}
\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] &= \nabla_\phi \int q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \int \frac{q_\phi(z|x_i)}{q_\phi(z|x_i)}\nabla_\phi q_\phi(z|x_i)\log p_\theta(x_i|z)\,dz \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\nabla_\phi \log q_\phi(z|x_i)\right] \\
&\approx \frac{1}{T}\log p_\theta(x_i|z^{(t)})\nabla_\phi \log q_\phi(z|x_i) \qquad\qquad z^{(t)} \sim q_\phi(z|x_i)
\end{aligned}
$$

High variance in practice

Variational auto-encoders (VAE)

- How to train a VAE?
- Assume Gaussian posterior

$$q(z|x) = \mathcal{N}\left(z; \mu_\phi(x), \Sigma_\phi(x)\right) \qquad \Sigma_\phi(x) = L_\phi(x) L_\phi(x)^T$$

- "Reparametrization trick"

$$\varepsilon \sim \mathcal{N}(\cdot; \mathbf{0}, \mathrm{I}) \qquad z = L_\phi(x)\varepsilon + \mu_\phi(x) \sim \mathcal{N}\left(\cdot; \mu_\phi(x), \Sigma_\phi(x)\right)$$

- 1st term

$$\mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] = \mathbb{E}_{\varepsilon\sim\mathcal{N}(\mathbf{0},\mathrm{I})}\left[\log p_\theta(x_i|L_\phi(x)\varepsilon + \mu_\phi(x))\right]$$

- Monte-Carlo estimate

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] \approx \frac{1}{T}\nabla_\phi \log p_\theta(x_i|L_\phi(x)\varepsilon^{(t)} + \mu_\phi(x)) \qquad \varepsilon^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$$

Works fine

Variational auto-encoders (VAE)

- How to train a VAE?
- Assume Gaussian posterior

$$q(z|x) = \mathcal{N}\big(z; \mu_\phi(x), \Sigma_\phi(x)\big) \qquad \Sigma_\phi(x) = L_\phi(x) L_\phi(x)^T$$

- "Reparametrization trick"

$$\varepsilon \sim \mathcal{N}(\cdot; \mathbf{0}, \mathrm{I}) \qquad z = L_\phi(x)\varepsilon + \mu_\phi(x) \sim \mathcal{N}\big(\cdot; \mu_\phi(x), \Sigma_\phi(x)\big)$$

- 1st term

$$\mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] = \mathbb{E}_{\varepsilon\sim\mathcal{N}(\mathbf{0},\mathrm{I})}\left[\log p_\theta(x_i|L_\phi(x)\varepsilon + \mu_\phi(x))\right]$$

- Monte-Carlo estimate

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] \approx \frac{1}{T}\nabla_\phi \log p_\theta(x_i|L_\phi(x)\varepsilon^{(t)} + \mu_\phi(x)) \qquad \varepsilon^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$$

Works fine

Variational auto-encoders (VAE)

- How to train a VAE?
- Assume Gaussian posterior

$$q(z|x) = \mathcal{N}\left(z; \mu_\phi(x), \Sigma_\phi(x)\right) \qquad \Sigma_\phi(x) = L_\phi(x) L_\phi(x)^T$$

- "Reparametrization trick"

$$\varepsilon \sim \mathcal{N}(\cdot; \mathbf{0}, \mathrm{I}) \qquad z = L_\phi(x)\varepsilon + \mu_\phi(x) \sim \mathcal{N}\left(\cdot; \mu_\phi(x), \Sigma_\phi(x)\right)$$

- 1st term

$$\mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] = \mathbb{E}_{\varepsilon\sim\mathcal{N}(\mathbf{0},\mathrm{I})}\left[\log p_\theta(x_i|L_\phi(x)\varepsilon + \mu_\phi(x))\right]$$

- Monte-Carlo estimate

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x_i)}\left[\log p_\theta(x_i|z)\right] \approx \frac{1}{T}\nabla_\phi \log p_\theta(x_i|L_\phi(x)\varepsilon^{(t)} + \mu_\phi(x)) \qquad \varepsilon^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$$

  Works fine

Variational auto-encoders (VAE)

- How to train a VAE?
- "Reparametrization trick": applicable
  - Location-scale family of distributions (e.g. Gaussian)
  - Easily invertible cdf: $\varepsilon \sim \mathcal{U}[0, 1]$
  - Functions of easier RVs
- Caveat: requires continuous latent variables
- Reparametrization trick needed for ∇_θ?

$$\nabla_\theta \mathbb{E}_{z \sim q_\phi(\cdot|x)} \left[\log p_\theta(x|z)\right] = \nabla_\theta \mathbb{E}_{\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathrm{I})} \left[\log p_\theta(x_i | L_\phi(x)\varepsilon + \mu_\phi(x))\right]$$

  - No difference (besides different parametrization of $\Sigma_\phi(x)$)
  - Drawing samples from $q_\phi(\cdot|x)$ is equivalent to samples $L_\phi(x)\varepsilon + \mu_\phi(x))$, $z \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$

Variational auto-encoders (VAE)

Connection to regular AE

- Deterministic encoder

$$z = f_\phi(x) \qquad\qquad q_\phi(z|x) = \delta_{f_\phi(x)}(z)$$

- Gaussian decoder

$$p_\theta(x|z) = \mathcal{N}(x; g_\theta(z), \sigma^2 \mathrm{I})$$

- Gaussian prior $p_z(z) = \mathcal{N}(z; \mathbf{0}, \mathrm{I})$

$$\begin{aligned}\log p_\theta(x) &\geq \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(\cdot|x)\|p_z\right) \\ &= -\frac{1}{2\sigma^2}\left\|x - g_\theta(f_\phi(x))\right\|^2 - \frac{1}{2}\left\|f_\theta(x)\right\|^2\end{aligned}$$

- Functions f_ϕ and g_θ can be deep networks

Single layer AEs

- We choose

$$f_\phi(x) = s(W^T x) \qquad g_\theta(z) = Wz \qquad p_\theta(x|z) = \mathcal{N}(x; g_\theta(z), \mathrm{I}) \qquad p_z(z) = \mathcal{N}(z; 0, \tau^2 \mathrm{I})$$

  with $\tau \gg 0$
- Recall $s(u) = S'(u)$

Score matching model

$$E(x) = -\log p(x) = \frac{1}{2}\|x\|^2 - \sum_k S(w_k^T x) + \log Z(W)$$

Training objective

$$\mathbb{E}_{x\sim p_d}\left[\frac{1}{2}\left\|x - Ws(W^T x)\right\|^2 + \frac{1}{2}\sum\nolimits_k \|w_k\|^2 s'(w_k^T x)\right] \to \min_W$$

Single layer AEs

- We choose

$$f_\phi(x) = s(W^T x) \qquad g_\theta(z) = Wz \qquad p_\theta(x|z) = \mathcal{N}(x; g_\theta(z), \mathrm{I}) \qquad p_z(z) = \mathcal{N}(z; 0, \tau^2 \mathrm{I})$$

  with $\tau \gg 0$
- Recall $s(u) = S'(u)$

Auto-encoding variational Bayes

$$E(x) = -\log p(x) \leq \frac{1}{2} \left\| x - Ws(W^T x) \right\|^2 + \frac{1}{2\tau^2} \left\| s(W^T x) \right\|^2$$

Training objective

$$\mathbb{E}_{x \sim p_d} \left[\frac{1}{2} \left\| x - Ws(W^T x) \right\|^2 + \frac{1}{2\tau^2} \sum\nolimits_k \left\| s(w_k^T x) \right\|^2 \right] \to \min_W$$

Single layer VAEs

- Encoder model for discrete hiddens $z_k \in \{0, 1\}$

$$q_\phi(z_k = 1|x) = \sigma(w_k^T x)$$

- Decoder

$$p_\theta(x|z) = \mathcal{N}(x; Wz, \mathrm{I}) \qquad \log p_\theta(x|z) \doteq -\frac{1}{2}\|x - Wz\|^2$$

- Lower bound

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(\cdot|x)\|p_z\right) \\ &\doteq -\frac{1}{2\sigma^2}\left\|x - W\sigma(W^T x)\right\|^2 - D_{KL}\left(q_\phi(\cdot|x)\|p_z\right)\end{aligned}$$

- $p_{z_k} = \mathrm{Ber}(1/2)$: $D_{KL}\left(q_\phi(\cdot|x)\|p_z\right) \doteq -\sum_k H(q_\phi(z_k|x))$

Auto-encoding variational Bayes: Bernoulli hiddens

Training objective

$$\mathbb{E}_{x\sim p_d}\left[\frac{1}{2}\left\|x - Ws(W^T x)\right\|^2 + \sum_k H(q_\phi(z_k|x))\right] \to \min_W$$

Outline

Variational NCE

Overview

Variational noise-contrastive estimation

- combines NCE with variational Bayes
- proposes a lower bound on the NCE objective
- allows unnormalized latent variable models

- Rhodes & Gutmann, “Variational Noise-Contrastive Estimation”

Variational NCE

- Recall the NCE objective

$$J_{NCE}(\theta) = \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- Now

$$p_\theta(x) = \int p_\theta(x,z)\,dz$$

- Naively plugging into J?

$$J_{NCE}(p_\theta) = \mathbb{E}_{x\sim p_d}\left[\log \frac{\int p_\theta(x,z)\,dz}{\int p_\theta(x,z)\,dz+\nu p_n(x)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{\int p_\theta(x,z)\,dz+\nu p_n(x)}\right]$$

- We cannot evaluate $\int p_\theta(x,z)\,dz$ or $\nabla_\theta \int p_\theta(x,z)\,dz$

Variational NCE

- Recall the NCE objective

$$J_{NCE}(\theta) = \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- Now

$$p_\theta(x) = \int p_\theta(x,z)\,dz$$

- Naively plugging into J?

$$J_{NCE}(p_\theta) = \mathbb{E}_{x\sim p_d}\left[\log \frac{\int p_\theta(x,z)\,dz}{\int p_\theta(x,z)\,dz+\nu p_n(x)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{\int p_\theta(x,z)\,dz+\nu p_n(x)}\right]$$

- We cannot evaluate $\int p_\theta(x,z)\,dz$ or $\nabla_\theta \int p_\theta(x,z)\,dz$

Variational NCE

- Focus on the first term in the NCE objective

$$J_1 := \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[\log \frac{1}{1+\frac{\nu p_n(x)}{p_\theta(x)}}\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]$$

- The mapping $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ with

$$g(r) = -\log\left(1+\frac{\nu}{r}\right)$$

is concave (why?)

Variational NCE

- Focus on the first term in the NCE objective

$$
\begin{aligned}
J_1 &:= \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[\log \frac{1}{1+\frac{\nu p_n(x)}{p_\theta(x)}}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]
\end{aligned}
$$

- The mapping $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ with

$$
g(r) = -\log\left(1+\frac{\nu}{r}\right)
$$

is concave (why?)

Variational NCE

- Focus on the first term in the NCE objective

$$
\begin{aligned}
J_1 &:= \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[\log \frac{1}{1+\frac{\nu p_n(x)}{p_\theta(x)}}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]
\end{aligned}
$$

- The mapping $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ with

$$
g(r) = -\log\left(1+\frac{\nu}{r}\right)
$$

is concave (why?)

Variational NCE

- Focus on the first term in the NCE objective

$$
\begin{aligned}
J_1 &:= \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[\log \frac{1}{1+\frac{\nu p_n(x)}{p_\theta(x)}}\right] \\
&= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]
\end{aligned}
$$

- The mapping $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ with

$$
g(r) = -\log\left(1+\frac{\nu}{r}\right)
$$

  is concave (why?)

Variational NCE

- Recall the 1st term

$$J_1 = \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] \qquad g(r) = -\log\left(1+\frac{\nu}{r}\right)$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] \qquad \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)}$$

$$\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]$$

Variational NCE

- Recall the 1st term

$$J_1 = \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] \qquad g(r) = -\log\left(1+\frac{\nu}{r}\right)$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right]$$

$$= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] \qquad \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)}$$

$$\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] \qquad \text{Jensen ineq.}$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]$$

Variational NCE

- Recall the 1st term

$$
\begin{aligned}
J_1 &= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] && g(r) = -\log\left(1+\frac{\nu}{r}\right) \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] && \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)} \\
&\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)\,g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] && \text{Jensen ineq.} \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]
\end{aligned}
$$

Variational NCE

- Recall the 1st term

$$
\begin{aligned}
J_1 &= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] && g(r) = -\log\left(1+\frac{\nu}{r}\right) \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] && \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)} \\
&\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)\,g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] && \text{Jensen ineq.} \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]
\end{aligned}
$$

Variational NCE

- Recall the 1st term

$$
\begin{aligned}
J_1 &= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] && g(r) = -\log\left(1+\frac{\nu}{r}\right) \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] && \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)} \\
&\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)\,g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] && \text{Jensen ineq.} \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]
\end{aligned}
$$

Variational NCE

- Recall the 1st term

$$
\begin{aligned}
J_1 &= \mathbb{E}_{x\sim p_d}\left[-\log\left(1+\frac{\nu p_n(x)}{p_\theta(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{p_\theta(x)}{p_n(x)}\right)\right] && g(r) = -\log\left(1+\frac{\nu}{r}\right) \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int p_\theta(x,z)\,dz\right)\right] \\
&= \mathbb{E}_{x\sim p_d}\left[g\left(\frac{1}{p_n(x)}\int \frac{q_\phi(z|x)p_\theta(x,z)}{q_\phi(x|z)}\,dz\right)\right] && \text{Insert } 1 = \frac{q_\phi(z|x)}{q_\phi(z|x)} \\
&\geq \mathbb{E}_{x\sim p_d}\left[\int q_\phi(z|x)\,g\left(\frac{1}{p_n(x)}\frac{p_\theta(x,z)}{q_\phi(x|z)}\right)\,dx\right] && \text{Jensen ineq.} \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]
\end{aligned}
$$

Variational NCE

- Recall

$$J_1 \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[-\log\left(1+\frac{\nu q_\phi(x|z)p_n(x)}{p_\theta(x,z)}\right)\right] \qquad \text{Expand } g$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right]$$

- Lower bound of the NCE objective

$$J_{NCE} \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- What should we do with the 2nd term?
  - Why does the same trick not work for the 2nd term?

Variational NCE

- Recall

$$J_1 \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right]$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[-\log\left(1+\frac{\nu q_\phi(x|z)p_n(x)}{p_\theta(x,z)}\right)\right] \qquad \text{Expand } g$$

$$= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right]$$

- Lower bound of the NCE objective

$$J_{NCE} \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- What should we do with the 2nd term?
  - Why does the same trick not work for the 2nd term?

Variational NCE

- Recall

$$
\begin{aligned}
J_1 &\geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[-\log\left(1+\frac{\nu q_\phi(x|z)p_n(x)}{p_\theta(x,z)}\right)\right] && \text{Expand } g \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right]
\end{aligned}
$$

- Lower bound of the NCE objective

$$
J_{NCE} \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]
$$

- What should we do with the 2nd term?
  - Why does the same trick not work for the 2nd term?

Variational NCE

- Recall

$$
\begin{aligned}
J_1 &\geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[g\left(\frac{p_\theta(x,z)}{q_\phi(x|z)p_n(x)}\right)\right] \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[-\log\left(1+\frac{\nu q_\phi(x|z)p_n(x)}{p_\theta(x,z)}\right)\right] && \text{Expand } g \\
&= \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right]
\end{aligned}
$$

- Lower bound of the NCE objective

$$
J_{NCE} \geq \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log\frac{p_\theta(x,z)}{\nu p_n(x)q_\phi(x|z)}\right] + \nu\mathbb{E}_{x\sim p_n}\left[\log\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]
$$

- What should we do with the 2nd term?
  - Why does the same trick not work for the 2nd term?

Variational NCE

- Recall lower bound

$$\mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- Paper proposes to use importance sampling for the 2nd terms

$$p_\theta(x) = \int p_\theta(x,z)\,dz = \int \frac{q_\phi(z|x)}{q_\phi(z|x)} p_\theta(x,z)\,dz = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

- We arrive at

$$J_{VNCE}(\theta,\phi) = \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right]$$

$$+ \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \nu p_n(x)}\right]$$

Variational NCE

- Recall lower bound

$$\mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- Paper proposes to use importance sampling for the 2nd terms

$$p_\theta(x) = \int p_\theta(x,z)\,dz = \int \frac{q_\phi(z|x)}{q_\phi(z|x)} p_\theta(x,z)\,dz = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

- We arrive at

$$J_{VNCE}(\theta,\phi) = \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right]$$

$$+ \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \nu p_n(x)}\right]$$

Variational NCE

- Recall lower bound

$$\mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$

- Paper proposes to use importance sampling for the 2nd terms

$$p_\theta(x) = \int p_\theta(x,z)\,dz = \int \frac{q_\phi(z|x)}{q_\phi(z|x)} p_\theta(x,z)\,dz = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

- We arrive at

$$J_{VNCE}(\theta,\phi) = \mathbb{E}_{x\sim p_d, z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{\nu p_n(x) q_\phi(x|z)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \nu p_n(x)}\right]$$

Variational NCE

- By construction we have

$$J_{VNCE}(\theta, \phi) \leq J_{NCE}(\theta)$$

- It can be shown that for a suitable f

$$J_{VNCE}(\theta, \phi) = J_{NCE}(\theta) - D_f\left(q_\phi(\cdot|x) \| p_\theta(\cdot|x)\right)$$

- $D_f(p\|q) \geq 0 \ldots f$-divergence (generalization of D_{KL})

$$D_f(p\|q) := \mathbb{E}_{x\sim q}\left[f\left(\frac{p(x)}{q(x)}\right)\right]$$

f is convex with $f(0) = 1$

$$D_{KL}(p\|q) = D_f(p\|q) \qquad \text{with } f(x) = x\log x$$

Discussion

- When is $J_{VNCE}(\theta, \phi) = J_{NCE}(\theta)$?
- What about unnormalized models?

Variational NCE

- By construction we have

$$J_{VNCE}(\theta, \phi) \leq J_{NCE}(\theta)$$

- It can be shown that for a suitable f

$$J_{VNCE}(\theta, \phi) = J_{NCE}(\theta) - D_f\left(q_\phi(\cdot|x) \| p_\theta(\cdot|x)\right)$$

- $D_f(p\|q) \geq 0 \ldots f$-divergence (generalization of D_{KL})

$$D_f(p\|q) := \mathbb{E}_{x\sim q}\left[f\left(\frac{p(x)}{q(x)}\right)\right]$$

f is convex with $f(0) = 1$

$$D_{KL}(p\|q) = D_f(p\|q) \qquad \text{with } f(x) = x\log x$$

Discussion

- When is $J_{VNCE}(\theta, \phi) = J_{NCE}(\theta)$?
- What about unnormalized models?

Outline

Boltzmann machine

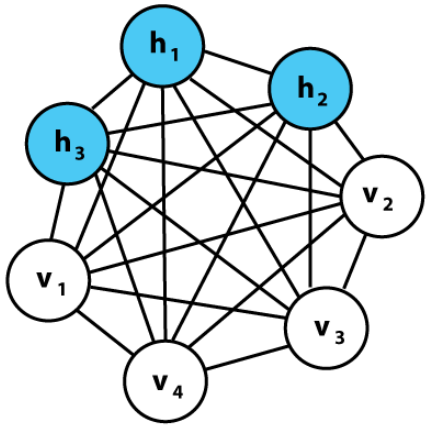
A Boltzmann machine is an energy-based model over binary variables $x_j \in \{0, 1\}$ with

$$\log p(x) = x^T W x + b^T x + c \qquad \text{diag}(W) = 0$$

- Boltzmann machine are undirected graphical models
- Strongly connected with Hopfield networks used to model associative memory
  - Clamp visible units $x_j, j \in V$ to given values

$$p(x_{\setminus V} | x_V) = ?$$

  - Intractable, since $x_{\setminus V}$ are not independent



Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is an energy-based model over two sets of binary variables $x_j \in \{0, 1\}$ and $z_k \in \{0, 1\}$ with

$$\log p(x, z) \doteq x^T W z + a^T x + b^T z$$

Visible units only talk to hidden units (and vice versa).

- Clamp hidden units z:

$$p(x|z) \propto \exp(x^T W z + a^T x) = \prod_j e^{x_j W_j z + a_j x_j} = \prod_j p(x_j|z)$$

$$p(x_j = 0|z) \propto e^0 = 1 \qquad p(x_j = 1|z) \propto e^{W_j z + a_j}$$

$$p(x_j = 1|z) = \frac{e^{W_j z + a_j}}{1 + e^{W_j z + a_j}} = \sigma(W_j z + a_j) \qquad \sigma(u) = \frac{e^u}{1 + e^u}$$

- Clamp visible units x:

$$p(z_k = 1|x) = \sigma(W^k x + b_k)$$

Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is an energy-based model over two sets of binary variables $x_j \in \{0, 1\}$ and $z_k \in \{0, 1\}$ with

$$\log p(x, z) \doteq x^T W z + a^T x + b^T z$$

Visible units only talk to hidden units (and vice versa).

- Clamp hidden units z:

$$p(x|z) \propto \exp(x^T W z + a^T x) = \prod_j e^{x_j W_j z + a_j x_j} = \prod_j p(x_j|z)$$

$$p(x_j = 0|z) \propto e^0 = 1 \qquad p(x_j = 1|z) \propto e^{W_j z + a_j}$$

$$p(x_j = 1|z) = \frac{e^{W_j z + a_j}}{1 + e^{W_j z + a_j}} = \sigma(W_j z + a_j) \qquad \sigma(u) = \frac{e^u}{1 + e^u}$$

- Clamp visible units x:

$$p(z_k = 1|x) = \sigma(W^k x + b_k)$$

Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is an energy-based model over two sets of binary variables $x_j \in \{0, 1\}$ and $z_k \in \{0, 1\}$ with

$$\log p(x, z) \doteq x^T W z + a^T x + b^T z$$

Visible units only talk to hidden units (and vice versa).

- Clamp hidden units z:

$$p(x|z) \propto \exp(x^T W z + a^T x) = \prod_j e^{x_j W_j z + a_j x_j} = \prod_j p(x_j|z)$$

$$p(x_j = 0|z) \propto e^0 = 1 \qquad p(x_j = 1|z) \propto e^{W_j z + a_j}$$

$$p(x_j = 1|z) = \frac{e^{W_j z + a_j}}{1 + e^{W_j z + a_j}} = \sigma(W_j z + a_j) \qquad \sigma(u) = \frac{e^u}{1 + e^u}$$

- Clamp visible units x:

$$p(z_k = 1|x) = \sigma(W^k x + b_k)$$

Restricted Boltzmann machines

- RBMs allow to compute $p(z|x)$ and $p(x|z)$ easily and *exactly*
  - Visible units x are conditionally independent given hidden units z
  - Hidden units z are are conditionally independent given visible units x
- RBMs naturally introduce layers
  - Deep Boltzmann machines are a stack of RBMs
  - Salakhutdinov & Hinton,"Deep boltzmann machines"
- Maximum likelihood training
  - Contrastive divergence and variants
  - Hinton, Training products of experts by minimizing contrastive divergence, 2002
  - We will point out possible different algorithms to train an RBM

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$p(x) = \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz$$

$$= \frac{1}{Z} \int \exp\left(\sum_k x^T W^k z_k + a^T x + b_k z_k\right)\, dz$$

$$= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k)$$

$$= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots$$

$$= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} (1 + \exp(x^T W^k + b_k))$$

- Therefore

$$\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$
\begin{aligned}
p(x) &= \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz \\
&= \frac{1}{Z} \int \exp\left(\sum_k x^T W^k z_k + a^T x + b_k z_k)\right)\, dz \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k) \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots \\
&= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} (1 + \exp(x^T W^k + b_k))
\end{aligned}
$$

- Therefore

$$
\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)
$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$
\begin{aligned}
p(x) &= \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz \\
&= \frac{1}{Z} \int \exp\left(\sum_k x^T W^k z_k + a^T x + b_k z_k)\right)\, dz \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k) \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots \\
&= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} (1 + \exp(x^T W^k + b_k))
\end{aligned}
$$

- Therefore

$$
\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)
$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$
\begin{aligned}
p(x) &= \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz \\
&= \frac{1}{Z} \int \exp\left(\sum_k x^T W^k z_k + a^T x + b_k z_k\right)\, dz \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k) \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots \\
&= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} (1 + \exp(x^T W^k + b_k))
\end{aligned}
$$

- Therefore

$$
\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)
$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$
\begin{aligned}
p(x) &= \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz \\
&= \frac{1}{Z} \int \exp\left(\sum\nolimits_k x^T W^k z_k + a^T x + b_k z_k)\right)\, dz \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k) \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots \\
&= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} \left(1 + \exp(x^T W^k + b_k)\right)
\end{aligned}
$$

- Therefore

$$
\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)
$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Restricted Boltzmann machines

We cannot compute $p(x, z)$ (or $p(x)$) exactly

- Partition function is still intractable
- But:

$$
\begin{aligned}
p(x) &= \frac{1}{Z} \int \exp(x^T W z + a^T x + b^T z)\, dz \\
&= \frac{1}{Z} \int \exp\left(\sum_k x^T W^k z_k + a^T x + b_k z_k\right)\, dz \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}, \ldots, z_K \in \{0,1\}} \prod_{k=1}^{K} \exp(x^T W^k z_k + b_k z_k) \\
&= \frac{\exp(a^T x)}{Z} \sum_{z_1 \in \{0,1\}} \exp(x^T W^1 z_1 + b_1 z_1) \sum_{z_2 \in \{0,1\}} \exp(x^T W^2 z_2 + b_2 z_2) \cdots \\
&= \frac{\exp(a^T x)}{Z} \prod_{k=1}^{K} \left(1 + \exp(x^T W^k + b_k)\right)
\end{aligned}
$$

- Therefore

$$
\log p(x) \doteq a^T x + \sum_k S(x^T W^k + b_k) \qquad S(u) := \log(1 + e^u) \qquad S'(u) = \sigma(u)
$$

- We can use NCE to train an RBM!
  - Score matching is not applicable as x is not continuous

Contrastive divergence

- RBMs (and other models) trained with contrastive divergence
- Gradient of MLE for unnormalized latent variable models

$$p_\theta(x) = \frac{\int e^{f_\theta(x,z)}\, dz}{Z(\theta)} \qquad Z(\theta) = \int e^{f_\theta(x,z)}\, dz\, dx$$

- What is

$$\begin{aligned}\nabla_\theta \log p_\theta(x) &= \nabla_\theta \log \frac{\int e^{f_\theta(x,z)}\, dz}{Z(\theta)} \\ &= \nabla_\theta \log \int e^{f_\theta(x,z)}\, dz - \nabla_\theta \log Z(\theta)?\end{aligned}$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\nabla_\theta \log Z(\theta) = \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz\, dx}{Z(\theta)}$$

$$= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx$$

$$= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\nabla_\theta \log Z(\theta) = \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)}$$

$$= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz\, dx}{Z(\theta)}$$

$$= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx$$

$$= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz\, dx}{Z(\theta)} \\
&= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx \\
&= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz\, dx}{Z(\theta)} \\
&= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx \\
&= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz\, dx}{Z(\theta)} \\
&= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx \\
&= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}$$

Contrastive divergence

- Let us start with the 2nd term (recall $p_\theta(x,z) = e^{f_\theta(x,z)}/Z(\theta)$)

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz\, dx}{Z(\theta)} \\
&= \frac{\int e^{f_\theta(x,z)} \left(\nabla_\theta \log e^{f_\theta(x,z)}\right)\, dz\, dx}{Z(\theta)} \\
&= \int p_\theta(x,z) \nabla_\theta f_\theta(x,z)\, dz\, dx \\
&= \mathbb{E}_{(x,z)\sim p_\theta(\cdot,\cdot)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}$$

Contrastive divergence

- Now for the 1st term

$$\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz = \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz}$$

$$= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)}$$

$$= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)}$$

$$= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)}$$

$$= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz}{p_\theta(x)}$$

$$= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right) dz$$

$$= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right) dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right) dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right) dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right) dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right) dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right)\, dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right)\, dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Now for the 1st term

$$
\begin{aligned}
\nabla_\theta \log \int e^{f_\theta(x,z)}\, dz &= \frac{\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{\int e^{f_\theta(x,z)}\, dz} \\
&= \frac{Z(\theta)\nabla_\theta \int e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{Z(\theta) \int \frac{e^{f_\theta(x,z)}}{e^{f_\theta(x,z)}} \nabla_\theta e^{f_\theta(x,z)}\, dz}{p_\theta(x)} \\
&= \frac{\int p_\theta(x,z) \left(\nabla_\theta \log e^{f_\theta(x,z)}\right)\, dz}{p_\theta(x)} \\
&= \int p_\theta(z|x) \left(\nabla_\theta f_\theta(x,z)\right)\, dz \\
&= \mathbb{E}_{z\sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right]
\end{aligned}
$$

Contrastive divergence

- Gradient of MLE for unnormalized latent variable models

$$p_\theta(x) = \frac{\int e^{f_\theta(x,z)}\, dz}{Z(\theta)} \qquad Z(\theta) = \int e^{f_\theta(x,z)}\, dz\, dx$$

- What is

$$\nabla_\theta \log p_\theta(x) = \nabla_\theta \log \int e^{f_\theta(x,z)}\, dz - \nabla_\theta \log Z(\theta)?$$

- Answer

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{z \sim p_\theta(\cdot|x)} \left[\nabla_\theta f_\theta(x,z)\right] - \mathbb{E}_{(x',z) \sim p_\theta} \left[\nabla_\theta f_\theta(x',z)\right]$$

- Expectation over the posterior minus expectation over the model

Contrastive divergence

Back to RBM

- Gradient of MLE for unnormalized latent variable models

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right] - \mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$$

- $\mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right]$ can be estimated from samples

$$p_\theta(z|x) = \prod_k Ber(z_k; \sigma(W^k x + b_k))$$

- $\mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$ much harder
  - We need to draw samples from the joint
  - Solution: Gibbs sampling

$$z \sim p_\theta(\cdot|x) \to x' \sim p_\theta(\cdot|z) \to z' \sim p_\theta(\cdot|x') \to x'' \sim p_\theta(\cdot|z') \to \cdots \to x^\infty \sim p_\theta(\cdot|z^\infty)$$

  - Contrastive divergence stops at (x', z) to produce a sample from p_θ

Contrastive divergence

Back to RBM

- Gradient of MLE for unnormalized latent variable models

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right] - \mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$$

- $\mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right]$ can be estimated from samples

$$p_\theta(z|x) = \prod_k Ber(z_k; \sigma(W^k x + b_k))$$

- $\mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$ much harder
  - We need to draw samples from the joint
  - Solution: Gibbs sampling

$$z \sim p_\theta(\cdot|x) \to x' \sim p_\theta(\cdot|z) \to z' \sim p_\theta(\cdot|x') \to x'' \sim p_\theta(\cdot|z') \to \cdots \to x^\infty \sim p_\theta(\cdot|z^\infty)$$

  - Contrastive divergence stops at (x', z) to produce a sample from p_θ

Contrastive divergence

Back to RBM

- Gradient of MLE for unnormalized latent variable models

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right] - \mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$$

- $\mathbb{E}_{z\sim p_\theta(\cdot|x)}\left[\nabla_\theta f_\theta(x,z)\right]$ can be estimated from samples

$$p_\theta(z|x) = \prod_k Ber(z_k; \sigma(W^k x + b_k))$$

- $\mathbb{E}_{(x',z)\sim p_\theta}\left[\nabla_\theta f_\theta(x',z)\right]$ much harder
  - We need to draw samples from the joint
  - Solution: Gibbs sampling

$$z \sim p_\theta(\cdot|x) \to x' \sim p_\theta(\cdot|z) \to z' \sim p_\theta(\cdot|x') \to x'' \sim p_\theta(\cdot|z') \to \cdots \to x^\infty \sim p_\theta(\cdot|z^\infty)$$

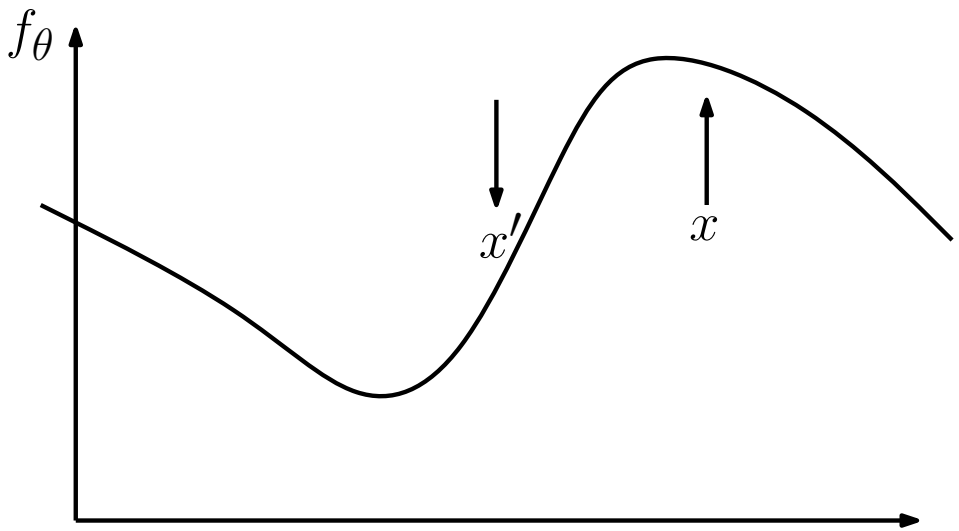
  - Contrastive divergence stops at (x', z) to produce a sample from p_θ

Contrastive divergence

Learning using energy-based models

Contrastive divergence learning increases (unnormalized) log-likelihood f_θ for given datapoints x and decreases it for nearby samples x'.

- Which other training methods share this feature?



Gaussian-Bernoulli RBMs (GRBMs)

Real-valued visible units?

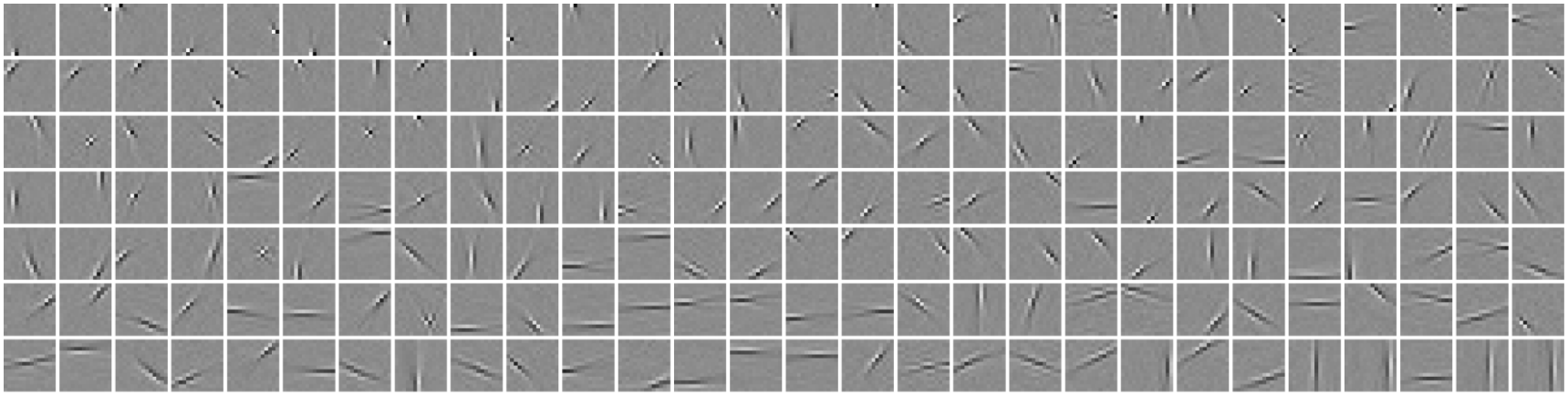
- Gaussian-Bernoulli RBM: $x_j \in \mathbb{R}$, $z_k \in \{0, 1\}$

$$\log p(x, z) \doteq -\frac{1}{2\sigma^2} \|x - a\|^2 + \frac{1}{\sigma^2} x^T W z + b^T z$$

- Hinton & Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks"

Questions

- What is $p(x|z)$?
- What is $p(x)$?



Melchior et al., "Gaussian-binary restricted Boltzmann machines for modeling natural image statistics"

Gaussian-Bernoulli RBMs (GRBMs)

Real-valued visible units?

- Gaussian-Bernoulli RBM: $x_j \in \mathbb{R}$, $z_k \in \{0, 1\}$

$$\log p(x, z) \doteq -\frac{1}{2\sigma^2} \|x - a\|^2 + \frac{1}{\sigma^2} x^T W z + b^T z$$

- Hinton & Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks"

Questions

- What is $p(x|z)$?

$$p(x|z) = \mathcal{N}(x; a + Wz, \sigma^2 \mathrm{I})$$

- What is $p(x)$?

$$\log p(x) \doteq -\frac{\|x - a\|^2}{2\sigma^2} + \sum\nolimits_k S(x^T W^k / \sigma^2 + b_k)$$

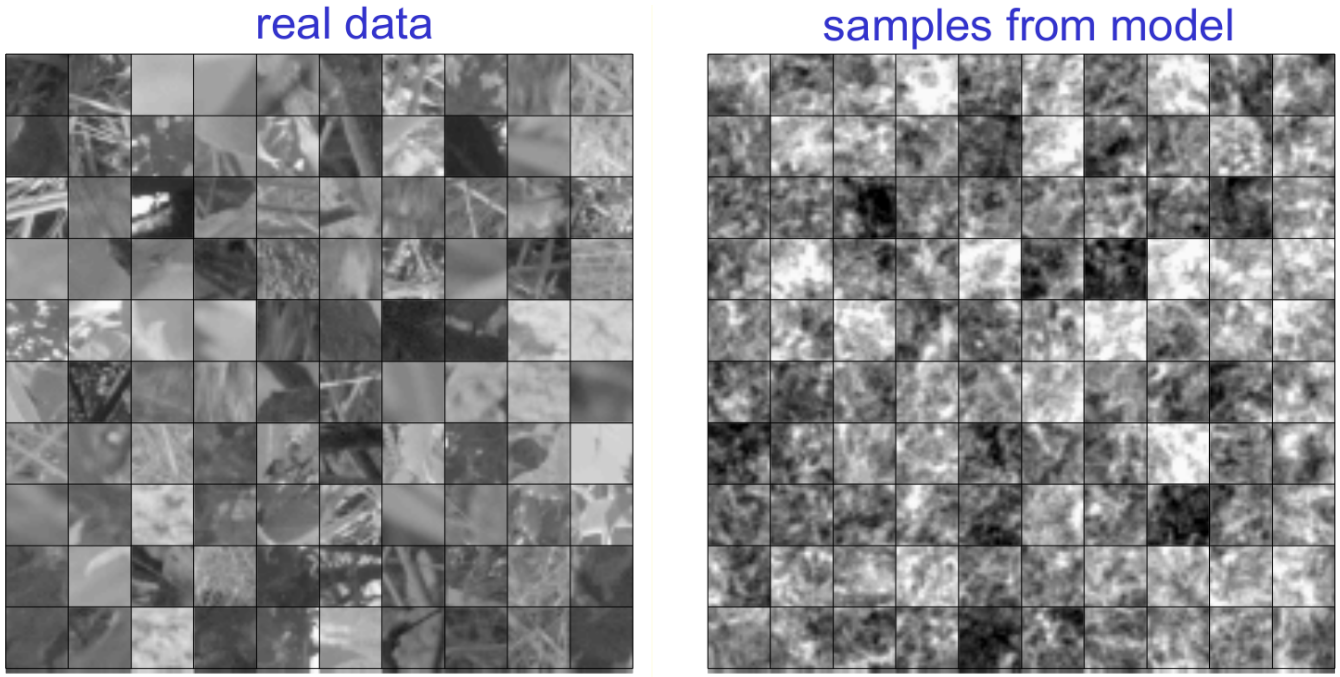
  Should look familar
- We can use score matching now as well!

Semi-restricted Boltzmann machines

- Semi-restricted Boltzmann machines allow coupling between visible units via lateral connections

$$\log p(x, z) \doteq -\frac{1}{2}(x - a)^T \Lambda (x - a) + b^T z + x^T \Lambda W z$$
$$p(x|z) = \mathcal{N}(x; a + Wz, \Lambda)$$

- Osindero & Hinton, “Modeling image patches with a directed hierarchy of Markov random fields”



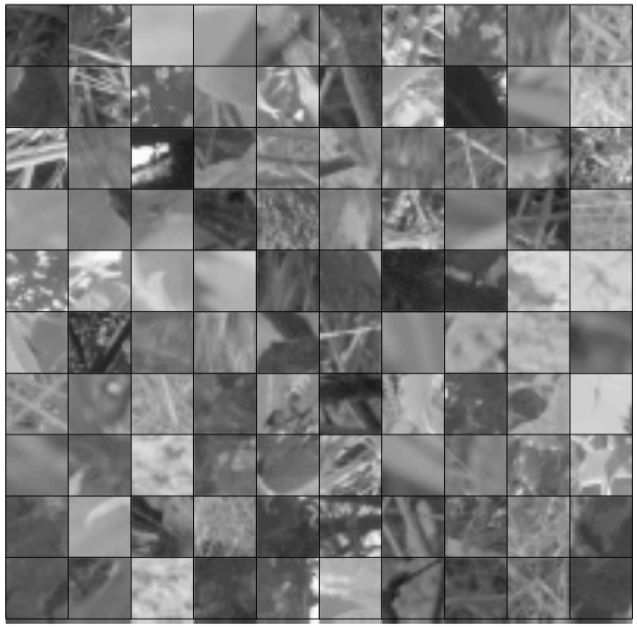
Semi-restricted Boltzmann machines

- Semi-restricted Boltzmann machines allow coupling between visible units via lateral connections

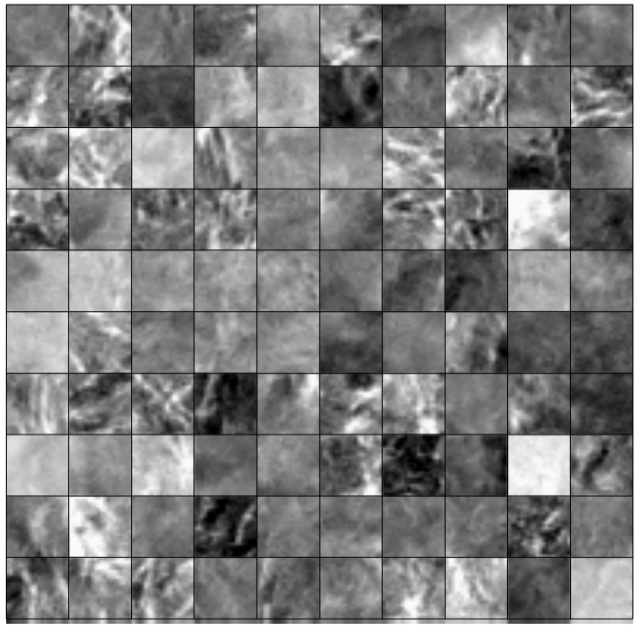
$$\log p(x, z) \doteq -\frac{1}{2}(x - a)^T \Lambda (x - a) + b^T z + x^T \Lambda W z$$
$$p(x|z) = \mathcal{N}(x; a + Wz, \Lambda)$$

- Osindero & Hinton, “Modeling image patches with a directed hierarchy of Markov random fields”

real data



samples from model



Mean-covariance RBM (mcRBM)

- Idea: two sets of hidden units
  - One set modulates the mean as in the GRBM
  - A 2nd set of hiddens modulates the covariance matrix
- Energy model $\log p(x, y, z) = \log p_M(x, z) + \log p_C(x, y)$

$$\log p_M(x, z) = \tfrac{1}{2} \|x - a\|^2 + x^T \mathsf{W} z + b^T x \qquad \text{GRBM}$$

$$\log p_C(x, y) = (x^T \mathsf{R})^2 \mathsf{P} y + c^T y \qquad \text{cRBM}$$

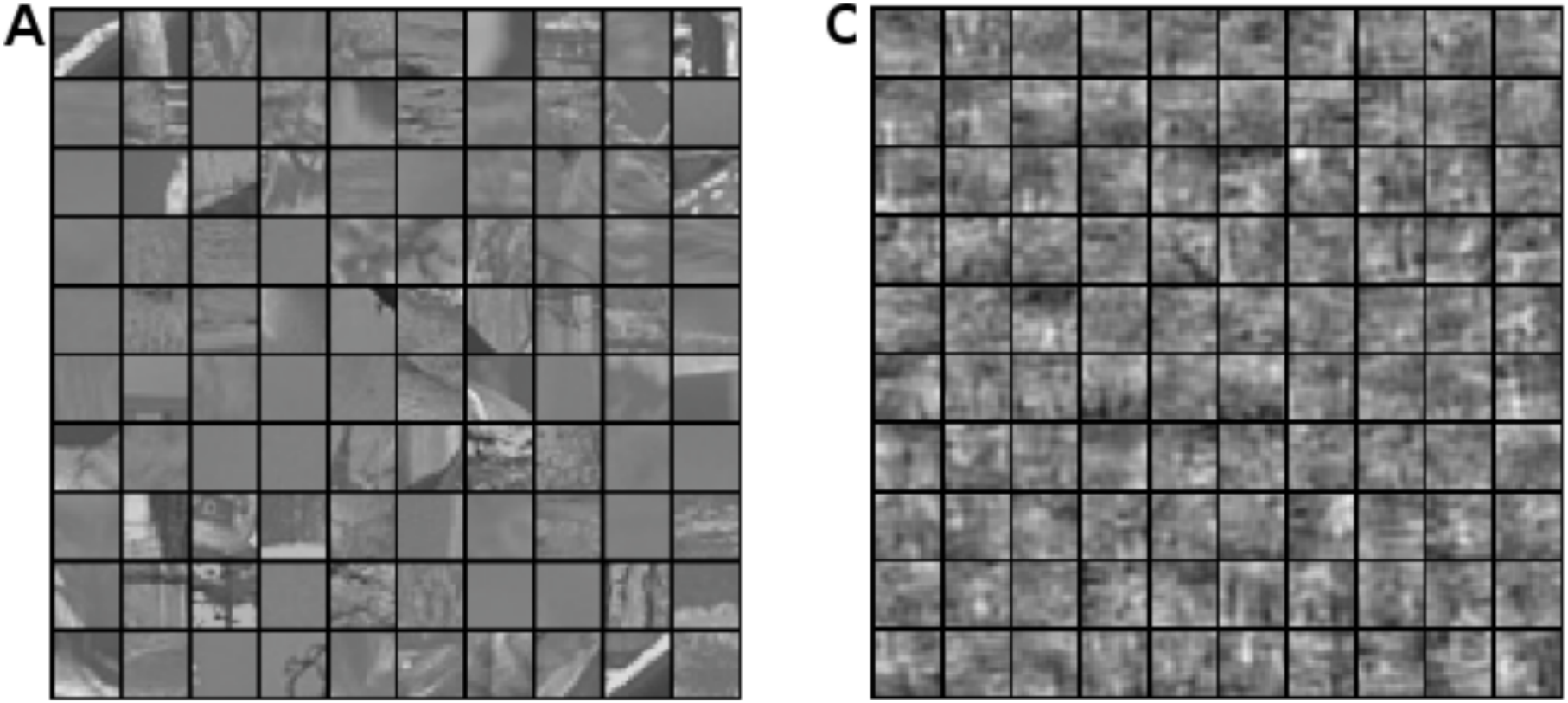
- It can be shown that

$$p(x|y, z) = \mathcal{N}(x; \Sigma \mathsf{W} z, \Sigma) \qquad \Sigma := \left(\mathsf{R}\, \mathrm{diag}(-\mathsf{P}^T y) \mathsf{R}^T\right)^{-1}$$

- Training using contrastive divergence
- Ranzato & Hinton, “Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines”

Mean-covariance RBM (mcRBM)

- Training on natural image patches

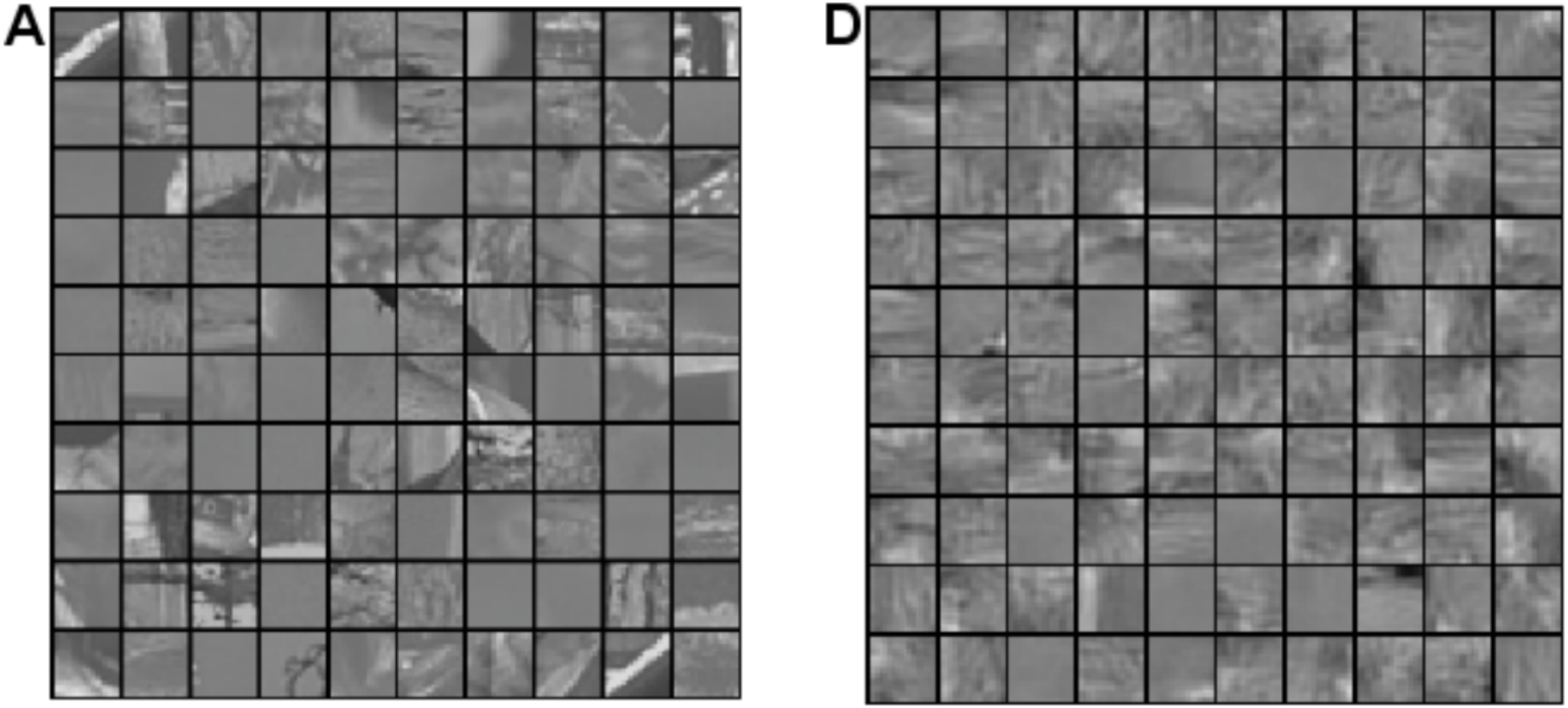


GRBM

- Ranzato & Hinton, “Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines”

Mean-covariance RBM (mcRBM)

- Training on natural image patches

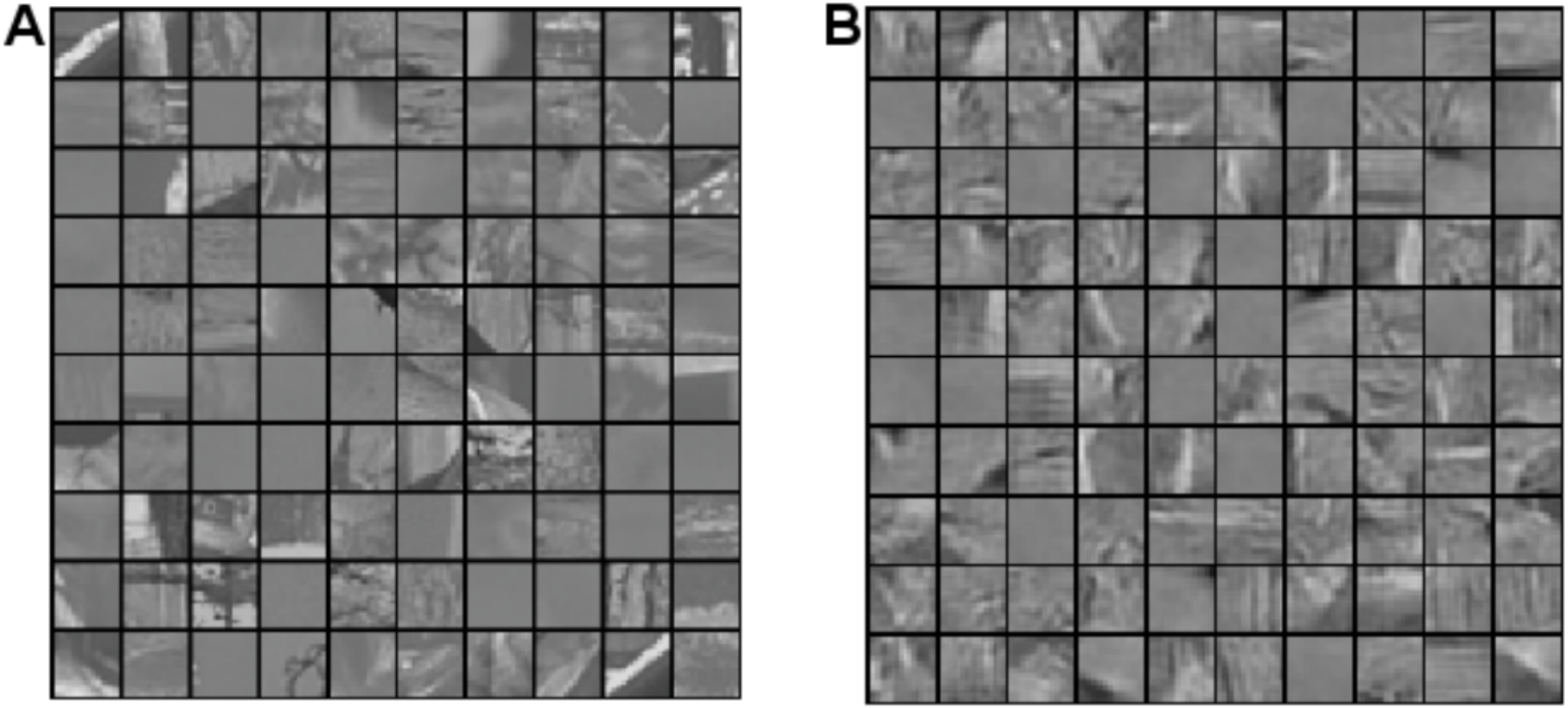


cRBM

- Ranzato & Hinton, “Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines”

Mean-covariance RBM (mcRBM)

- Training on natural image patches



mcRBM

- Ranzato & Hinton, “Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines”

Higher-order Boltzmann machines

- mcRBM is 3rd order Boltzmann machine
  - 3-way multiplicative interaction $z_k x_j x_{j'}$
- What about 3-way interaction $z_k x_j y_n$? Learning image relations

$$\log p(x, y, z) \doteq \sum_k z_k \, x^T W_k y + b^T x + c^T y + d^T z - \frac{1}{2} \|x\|^2 - \frac{1}{2} \|y\|^2$$

$$p(x|y,z) = \mathcal{N}\left(x; b + \left(\sum\nolimits_k z_k W_k\right) y, \mathrm{I}\right)$$

$$p(y|x,z) = \mathcal{N}\left(y; c + \left(\sum\nolimits_k z_k W_k^T\right) x, \mathrm{I}\right)$$

$$p(z|x,y) = \prod_k \mathrm{Ber}\left(z_k; \sigma\left(x^T W_k y + d_k\right)\right)$$

- Hidden units z determine the linear transformation
- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- mcRBM is 3rd order Boltzmann machine
  - 3-way multiplicative interaction $z_k x_j x_{j'}$
- What about 3-way interaction $z_k x_j y_n$? Learning image relations

$$\log p(x, y, z) \doteq \sum_k z_k \, x^T W_k y + b^T x + c^T y + d^T z - \frac{1}{2} \|x\|^2 - \frac{1}{2} \|y\|^2$$

$$p(x|y, z) = \mathcal{N}\left(x; b + \left(\sum\nolimits_k z_k W_k\right) y, \mathsf{I}\right)$$

$$p(y|x, z) = \mathcal{N}\left(y; c + \left(\sum\nolimits_k z_k W_k^T\right) x, \mathsf{I}\right)$$

$$p(z|x, y) = \prod_k \mathrm{Ber}\left(z_k; \sigma\left(x^T W_k y + d_k\right)\right)$$

- Hidden units z determine the linear transformation
- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- Learning image relations

$$\log p(x, y, z) \doteq \sum_k z_k \, x^T W_k y + b^T x + c^T y + d^T z - \frac{1}{2} \|x\|^2 - \frac{1}{2} \|y\|^2$$

- Efficiency?
  - $\mathcal{W} := (W_k)_{k=1}^N$ forms a 3-tensor
  - Many parameters $D \times D \times K$
  - Replace by rank-F approximation

$$\mathcal{W} = \sum_{f=1}^{F} w_f^x \otimes w_f^y \otimes w_f^z$$

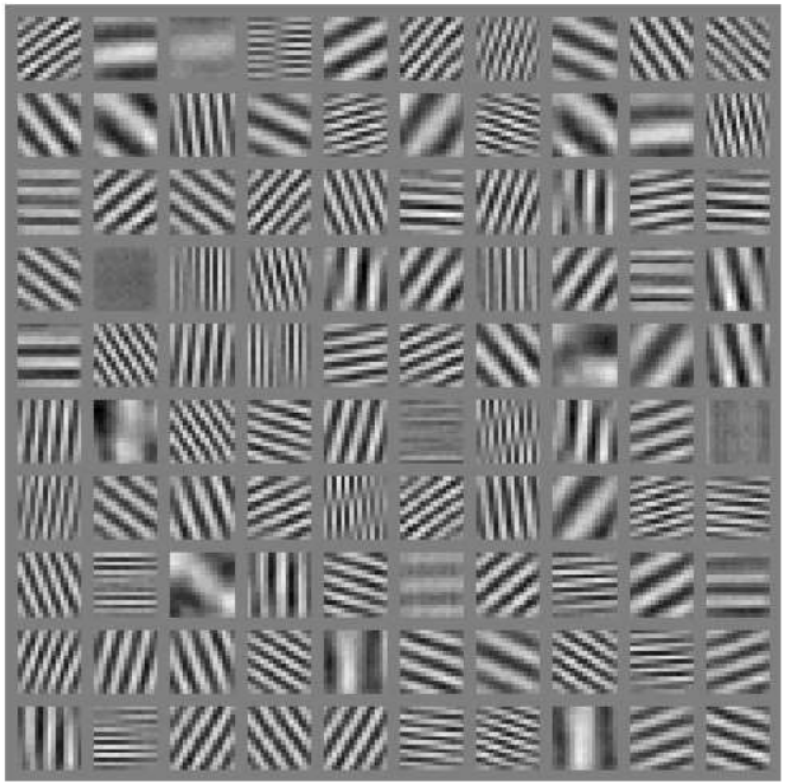
  - *Factored* higher-order RBM

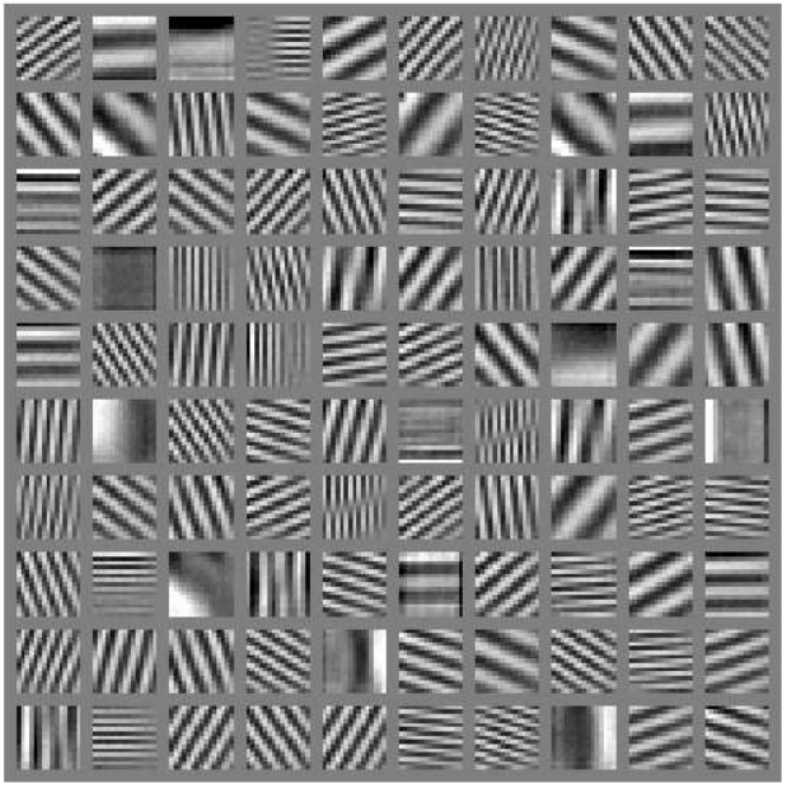
$$\log p(x, y, z) \doteq \sum_{k,f} w_{fk}^z z_k (x^T w_f^x)(y^T w_f^y) + b^T x + c^T y + d^T z - \frac{1}{2} \|x\|^2 - \frac{1}{2} \|y\|^2$$

- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- Learning: training set of corresponding image pairs $\{(x_i, y_i)\}$
- Filters w_f^x and w_f^y learned from data
  - Filters independent of image content
  - translated pairs

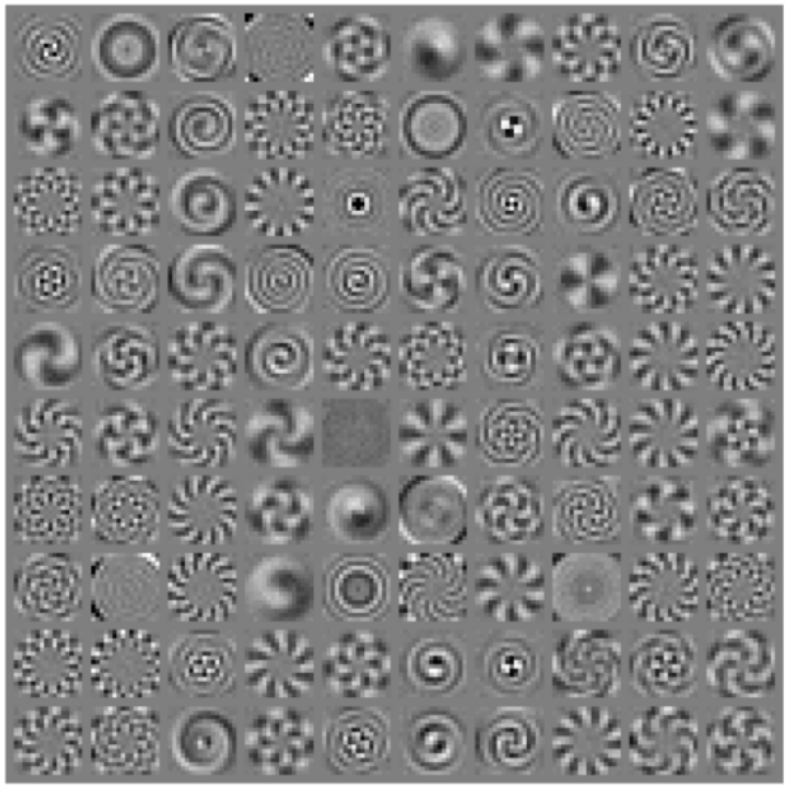


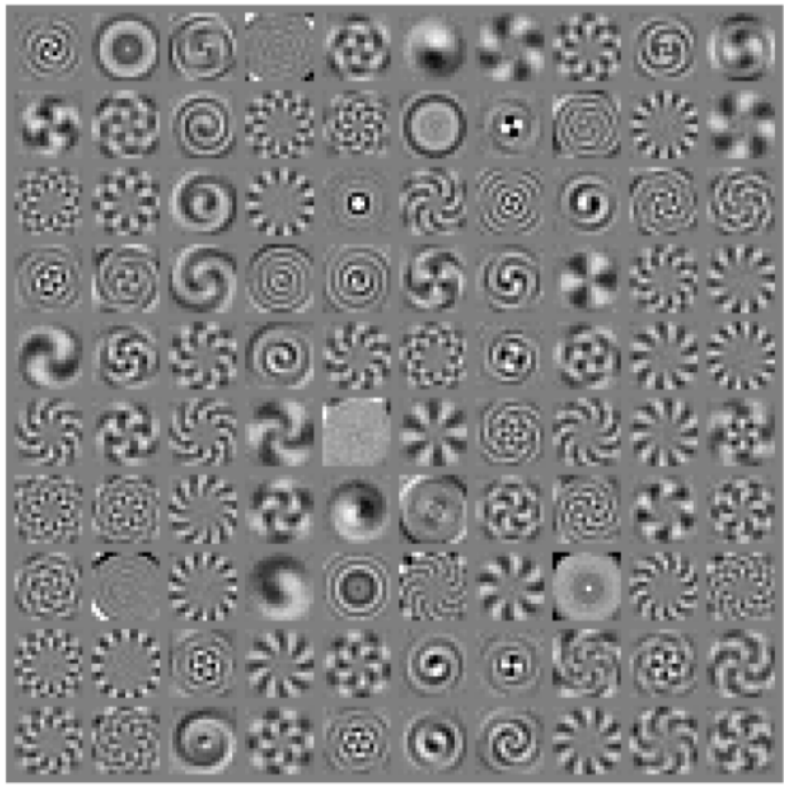


- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- Learning: training set of corresponding image pairs $\{(x_i, y_i)\}$
- Filters w_f^x and w_f^y learned from data
  - Filters independent of image content
  - rotated pairs

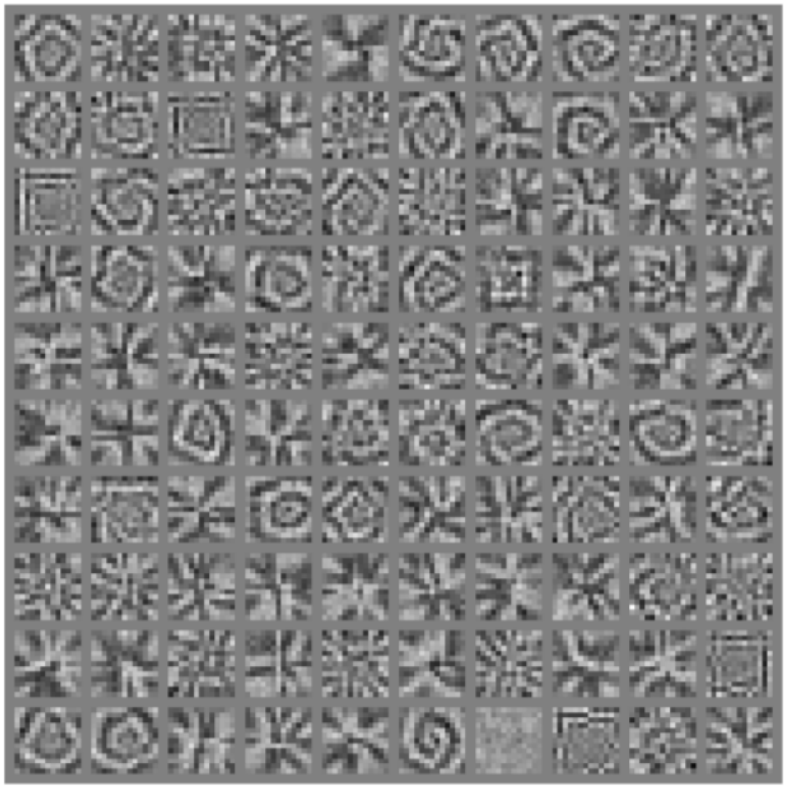


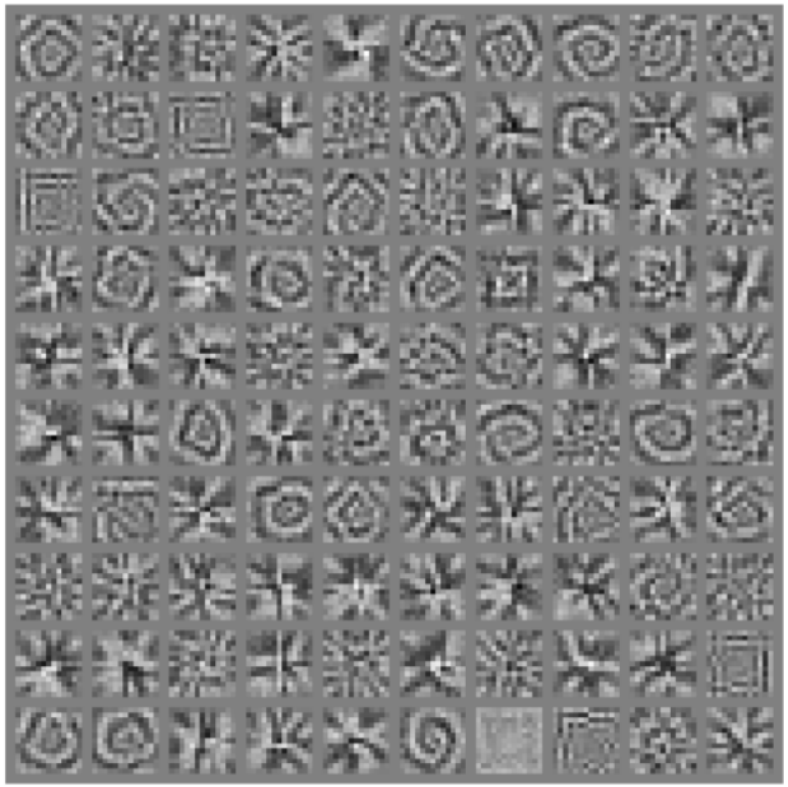


- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- Learning: training set of corresponding image pairs $\{(x_i, y_i)\}$
- Filters w_f^x and w_f^y learned from data
  - Filters independent of image content
  - scaled pairs





- Memisevic & Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines”

Higher-order Boltzmann machines

- Theory predicts how w_f^x and w_f^y should look
- Assume transformations (parametrized by κ) are orthonormal

$$x = A_\kappa y \qquad\qquad y = A_\kappa^T x$$

  - Permutations, rotations, translations (with wrap-around boundary conditions)
  - Eigen-decomposition: complex eigenvalues with magnitude 1

$$A_\kappa = U_\kappa \Lambda_\kappa U_\kappa^T$$

- Assume transformations commute: $A_\kappa A_{\kappa'} = A_{\kappa'} A_\kappa$
  - Commuting matrices share eigenspaces!

$$U_\kappa = U_{\kappa'} = U$$

- We obtain

$$x^T A_\theta y = (\underbrace{U^T x}_{x^T w^x})^T \underbrace{\Lambda_\theta}_{\mathrm{diag}(w^z z_k)} \underbrace{U^T y}_{y^T w^y}$$

- z parametrizes transformation κ
- Memisevic, “Learning related images”

Higher-order Boltzmann machines

- Theory predicts how w_f^x and w_f^y should look
- Assume transformations (parametrized by κ) are orthonormal

$$x = A_\kappa y \qquad\qquad y = A_\kappa^T x$$

  - Permutations, rotations, translations (with wrap-around boundary conditions)
  - Eigen-decomposition: complex eigenvalues with magnitude 1

$$A_\kappa = U_\kappa \Lambda_\kappa U_\kappa^T$$

- Assume transformations commute: $A_\kappa A_{\kappa'} = A_{\kappa'} A_\kappa$
  - Commuting matrices share eigenspaces!

$$U_\kappa = U_{\kappa'} = U$$

- We obtain

$$x^T A_\theta y = (\underbrace{U^T x}_{x^T w^x})^T \underbrace{\Lambda_\theta}_{\mathrm{diag}(w^z z_k)} \underbrace{U^T y}_{y^T w^y}$$

- z parametrizes transformation κ
- Memisevic, “Learning related images”

Higher-order Boltzmann machines

- Theory predicts how w_f^x and w_f^y should look
- Assume transformations (parametrized by κ) are orthonormal

$$x = A_\kappa y \qquad\qquad y = A_\kappa^T x$$

  - Permutations, rotations, translations (with wrap-around boundary conditions)
  - Eigen-decomposition: complex eigenvalues with magnitude 1

$$A_\kappa = U_\kappa \Lambda_\kappa U_\kappa^T$$

- Assume transformations commute: $A_\kappa A_{\kappa'} = A_{\kappa'} A_\kappa$
  - Commuting matrices share eigenspaces!

$$U_\kappa = U_{\kappa'} = U$$

- We obtain

$$x^T A_\theta y = (\underbrace{U^T x}_{x^T w^x})^T \underbrace{\Lambda_\theta}_{\mathrm{diag}(w^z z_k)} \underbrace{U^T y}_{y^T w^y}$$

- z parametrizes transformation κ
- Memisevic, “Learning related images”

Training RBMs and whitening of data

- Pixels in images are highly correlated
- Whitening aims to remove 2nd order correlation from images
  - Learned model can focus on higher-order correlations
- PCA whitening
  - Subtract empirical mean

$$\mathsf{X} \leftarrow \mathsf{X} - \tfrac{1}{N}\mathsf{X}\mathbf{1} \qquad\qquad \mathsf{X} = [x_1, \ldots, x_N]$$

  - Empirical covariance matrix and eigendecomposition

$$\mathsf{C} \leftarrow \tfrac{1}{N-1}\mathsf{X}\mathsf{X}^T \qquad\qquad \mathsf{C} = \mathsf{U}\Lambda\mathsf{U}^T$$

  - Whitening step

$$\mathsf{X} \leftarrow \Lambda^{-1/2}\mathsf{U}^T\mathsf{X}$$

- Krizhevsky, “Learning Multiple Layers of Features from Tiny Images”

Training RBMs and whitening of data

- Pixels in images are highly correlated
- Whitening aims to remove 2nd order correlation from images
  - Learned model can focus on higher-order correlations
- ZCA whitening
  - Subtract empirical mean

$$\mathsf{X} \leftarrow \mathsf{X} - \tfrac{1}{N}\mathsf{X}\mathbf{1} \qquad\qquad \mathsf{X} = [x_1, \ldots, x_N]$$

  - Empirical covariance matrix and eigendecomposition

$$\mathsf{C} \leftarrow \tfrac{1}{N-1}\mathsf{X}\mathsf{X}^T \qquad\qquad \mathsf{C} = \mathsf{U}\Lambda\mathsf{U}^T$$

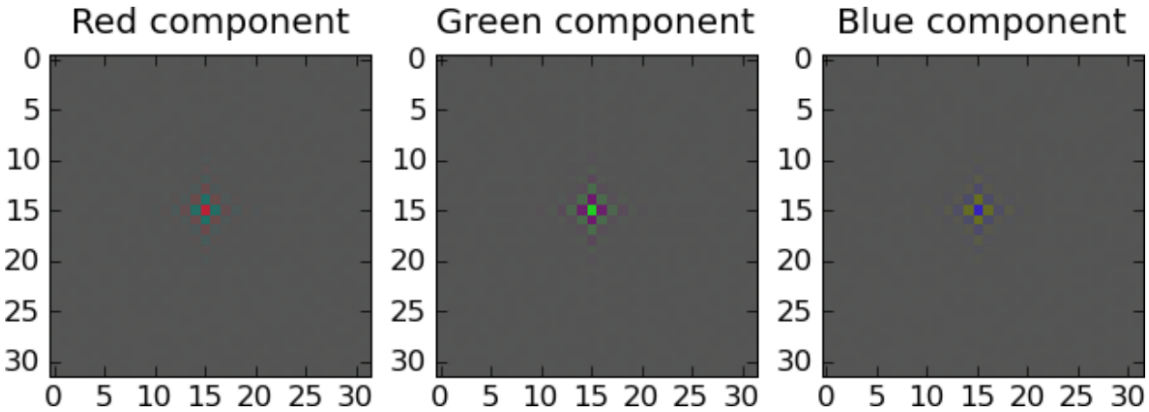
  - Whitening step: also rotate back to original space

$$\mathsf{X} \leftarrow \mathsf{U}^T\Lambda^{-1/2}\mathsf{U}^T\mathsf{X}$$

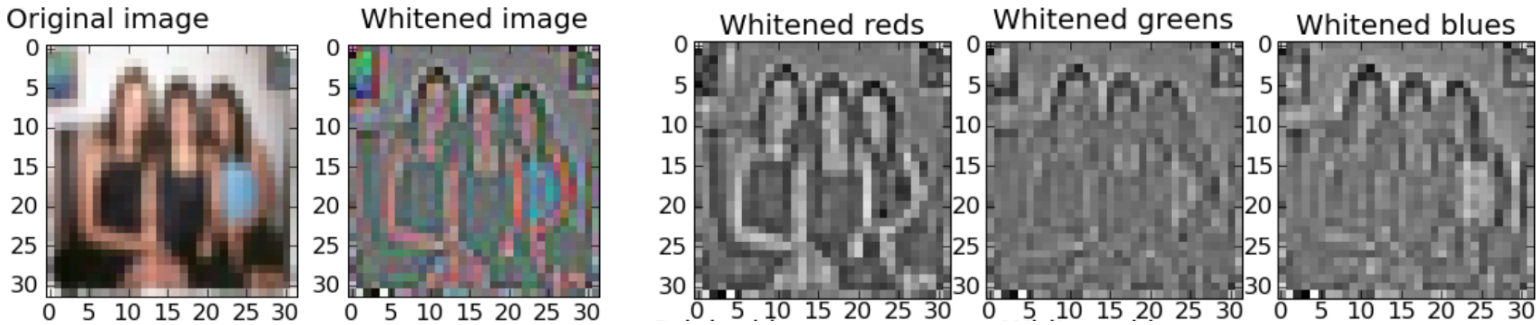
- Krizhevsky, “Learning Multiple Layers of Features from Tiny Images”

Training RBMs and whitening of data

- Whitening filters $U^T \Lambda^{-1/2} U^T$ for tiny images



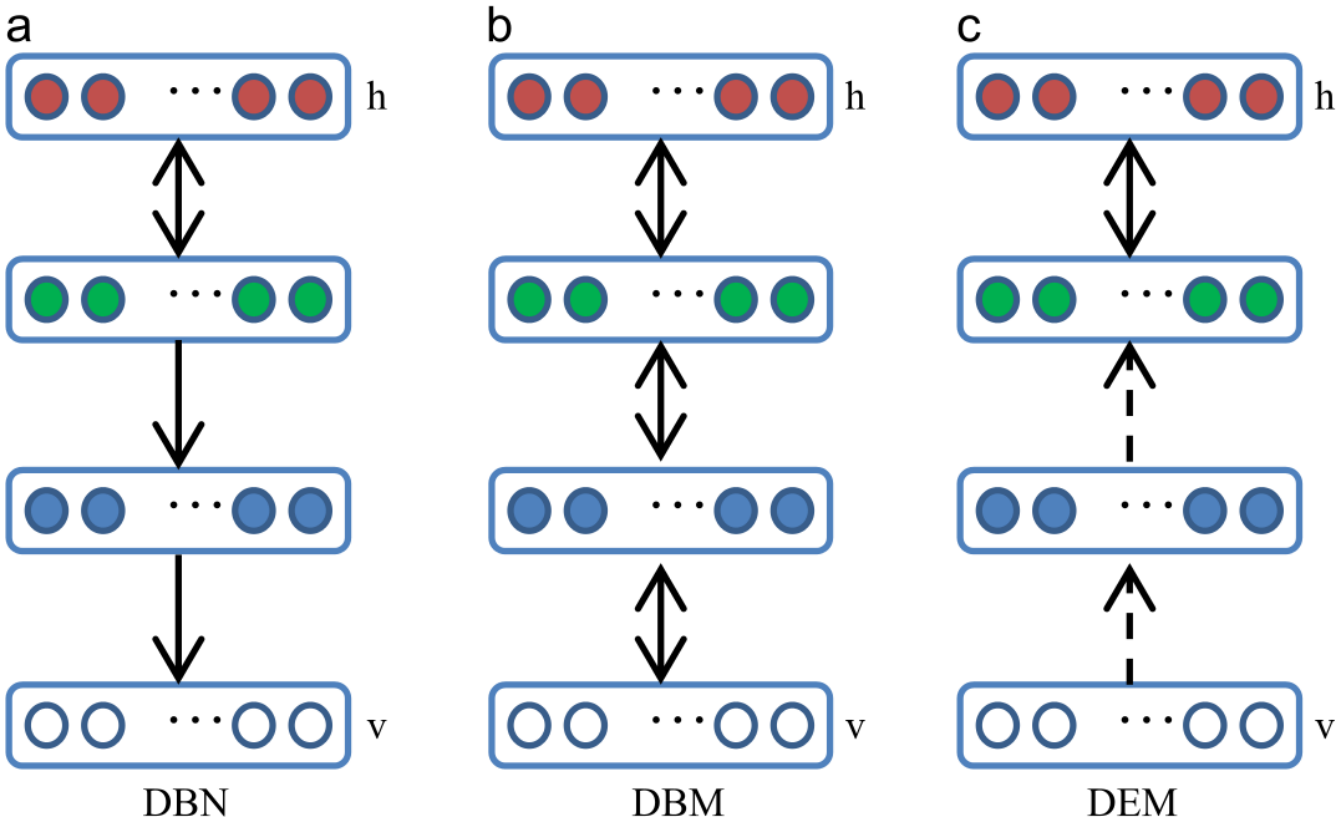
- ZCA whitening enhances edges in images



- Reason for observing Gabor filters in the 1st layer?
- PCA whitening leads to unrecognizable images
- Krizhevsky, “Learning Multiple Layers of Features from Tiny Images”

Deep extensions of RBMs

- Deep belief networks (Hinton et al., “A Fast Learning Algorithm for Deep Belief Nets”, 2006)
- Deep Boltzmann machines (Salakhutdinov & Hinton, “Deep Boltzmann machines”, 2009)
- Deep energy models (Ngiam et al., “Learning Deep Energy Models”, 2011)



Deep energy models (DEMs)

- Deep energy model (DEM)

$$\log p_\theta(x, z) \doteq -\frac{1}{2\sigma^2}\|x\|^2 + b^T x + c^T z + z^T W g_\theta(x) \qquad z \in \{0, 1\}^K$$

- $-\|x\|^2$ makes the model (usually) normalizable
- Marginalizing over z

$$p_\theta(z_k = 1|x) = s\left(w_k^T g_\theta(x) + c_k\right)$$

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2}\|x\|^2 + b^T x + \sum_k S\left(w_k^T g_\theta(x) + c_k\right)$$

- $p_\theta(x|z)$ is difficult in general
- General DEM

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2}\|x\|^2 + b^T x + H_\theta(x)$$

- Generalization of
  - GRBMs
  - mcRBMs
- Ngiam et al., “Learning Deep Energy Models”

Deep energy models (DEMs)

- Deep energy model (DEM)

$$\log p_\theta(x, z) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + c^T z + z^T W g_\theta(x) \qquad z \in \{0, 1\}^K$$

- $-\|x\|^2$ makes the model (usually) normalizable
- Marginalizing over z

$$p_\theta(z_k = 1|x) = s\left(w_k^T g_\theta(x) + c_k\right)$$

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + \sum_k S\left(w_k^T g_\theta(x) + c_k\right)$$

- $p_\theta(x|z)$ is difficult in general
- General DEM

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + H_\theta(x)$$

- Generalization of
  - GRBMs
  - mcRBMs
- Ngiam et al., “Learning Deep Energy Models”

Deep energy models (DEMs)

- Deep energy model (DEM)

$$\log p_\theta(x, z) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + c^T z + z^T W g_\theta(x) \qquad z \in \{0, 1\}^K$$

- $-\|x\|^2$ makes the model (usually) normalizable
- Marginalizing over z

$$p_\theta(z_k = 1|x) = s\left(w_k^T g_\theta(x) + c_k\right)$$

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + \sum_k S\left(w_k^T g_\theta(x) + c_k\right)$$

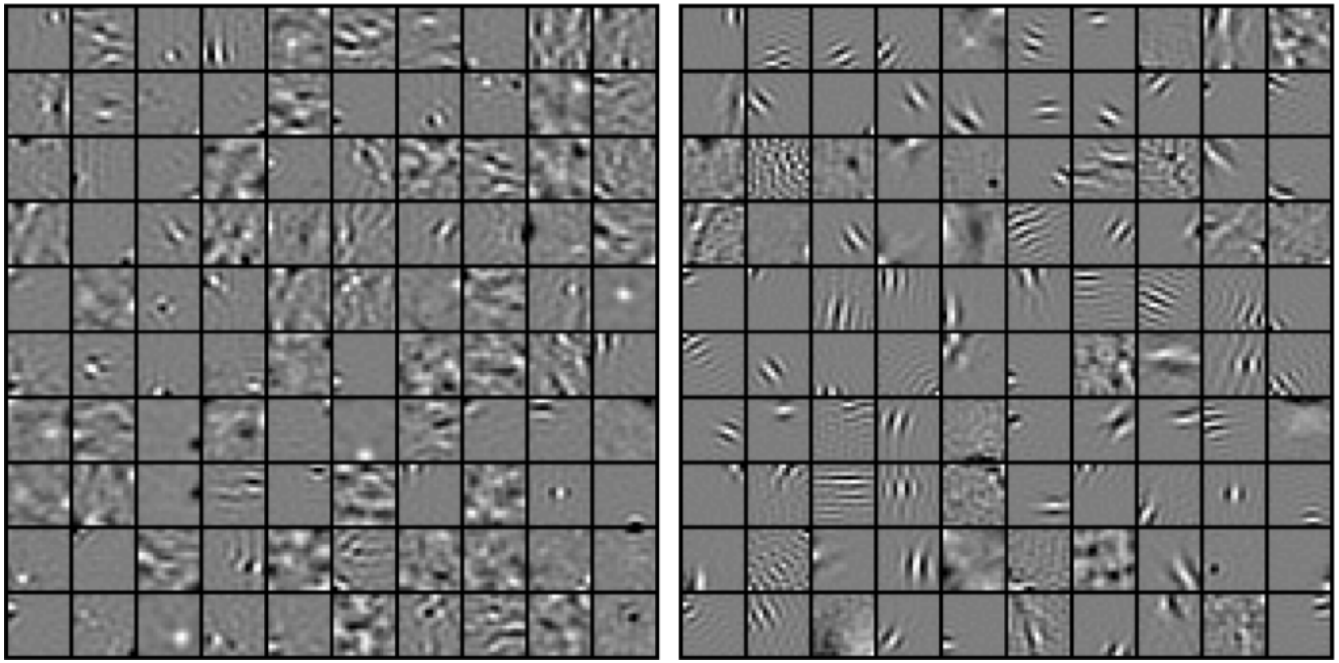
- $p_\theta(x|z)$ is difficult in general
- General DEM

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2} \|x\|^2 + b^T x + H_\theta(x)$$

- Generalization of
  - GRBMs
  - mcRBMs
- Ngiam et al., “Learning Deep Energy Models”

Deep energy models (DEMs)

- Learning DEM via contrastive divergence
  - Ngiam et al., “Learning Deep Energy Models”, 2011

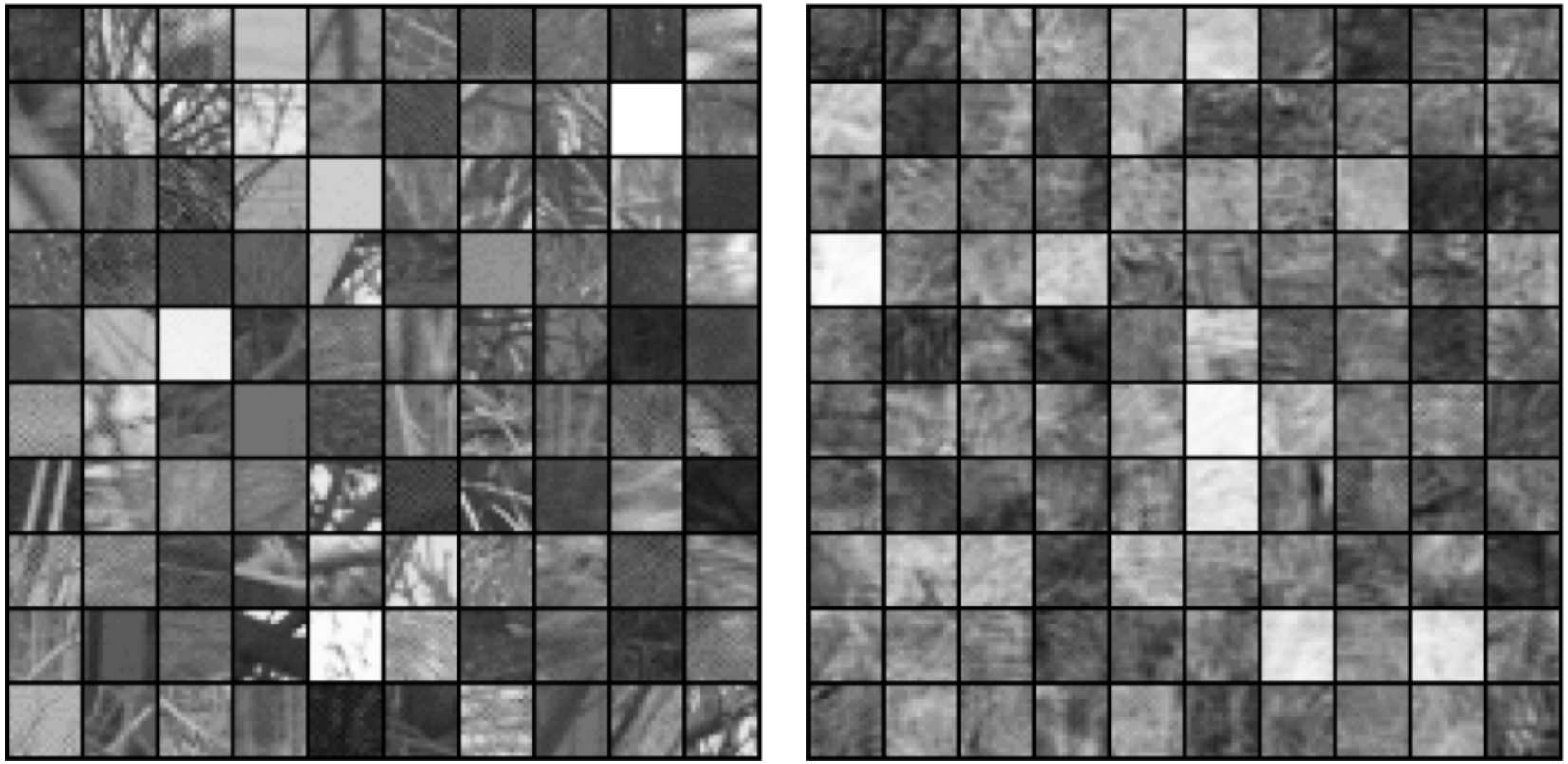


Single layer (sigmoid) DEM vs. after adding a 2nd layer

- Learning similar DEMs via score matching
  - Köster & Hyvärinen, “A Two-Layer Model of Natural Stimuli Estimated with Score Matching”, 2009
  - Kingma & LeCun, “Regularized estimation of image statistics by Score Matching”, 2010

Deep energy models (DEMs)

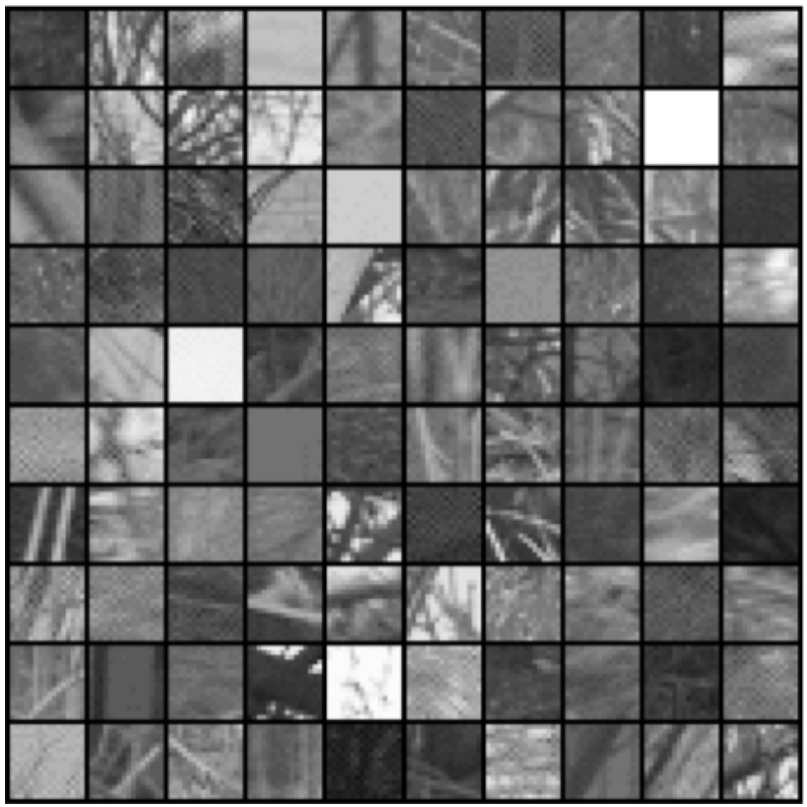
- Left: samples from training data
- Right: samples from model with 1 layer(s) in H_θ

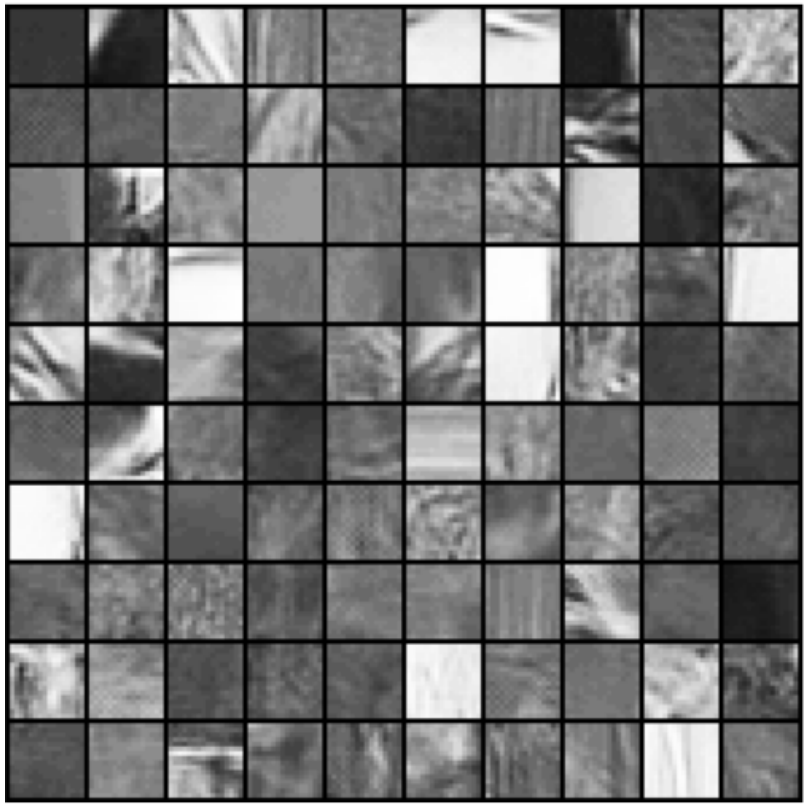


- Ngiam et al., “Learning Deep Energy Models”

Deep energy models (DEMs)

- Left: samples from training data
- Right: samples from model with 2 layer(s) in H_θ

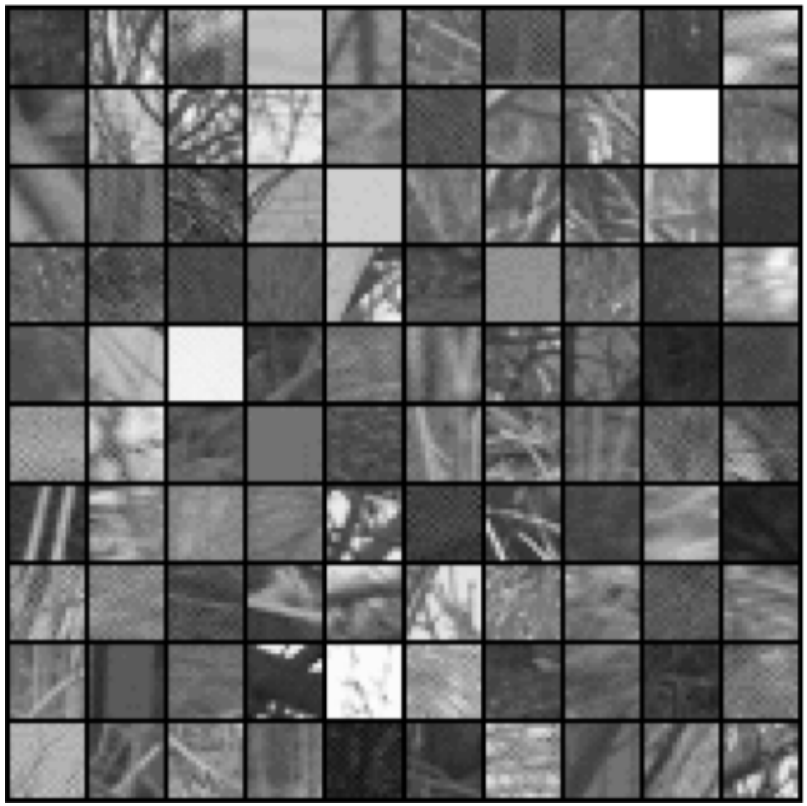


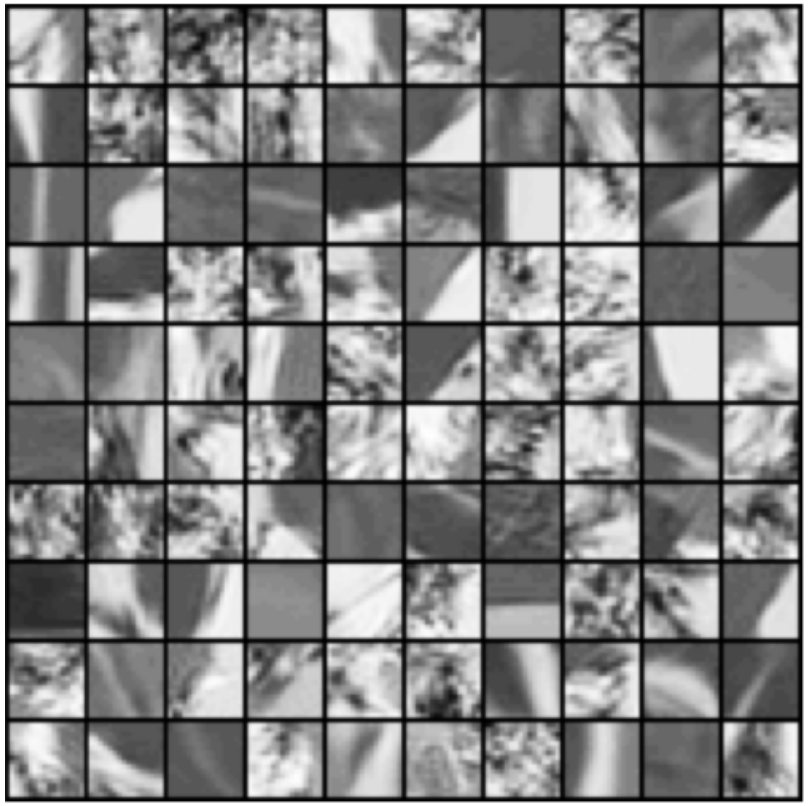


- Ngiam et al., “Learning Deep Energy Models”

Deep energy models (DEMs)

- Left: samples from training data
- Right: samples from model with 3 layer(s) in H_θ





- Ngiam et al., “Learning Deep Energy Models”

Outline

Sparse coding and dictionary learning

Overview

Sparse coding

- represents an input signal as sparse linear combination of atoms (dictionary elements)
- uses overcomplete dictionaries
- appears in
  - signal processing (wavelets)
  - statistics (LASSO)
  - compressed sensing
  - neuroscience

- Tibshirani, “Regression Shrinkage and Selection via the LASSO”
- Donoho & Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization”
- Olshausen & Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”

Sparse coding and dictionary learning

- Main objective

$$J(\mathsf{D}, \mathbf{z}) = \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q \qquad q \in [0, 1]$$

- Dictionary D
  - Columns of D are dictionary elements / atoms
  - Usually #rows(D) $\ll$ #columns(D)
- Sparse code $\mathbf{z}$: sparsity favored via $\|\mathbf{z}\|_q$
- Sparse coding: given D and $\mathbf{x}$ infer

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q$$

- Dictionary learning: given $\{x_i\}$ infer D s.t. $\|\mathsf{D}(:, j)\| \leq 1$ (why?)

$$\min_{\mathsf{D}} \sum_i \left(\min_{\mathbf{z}_i} \frac{1}{2} \|\mathbf{x}_i - \mathsf{D}\mathbf{z}_i\|^2 + \lambda \|\mathbf{z}_i\|_q \right)$$

- Usually not interpreted as probabilistic model
  - But: Bayesian sparse learning and automatic relevance determination

Sparse coding and dictionary learning

- Main objective

$$J(\mathsf{D}, \mathbf{z}) = \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q \qquad q \in [0, 1]$$

- Dictionary D
  - Columns of D are dictionary elements / atoms
  - Usually #rows(D) $\ll$ #columns(D)
- Sparse code $\mathbf{z}$: sparsity favored via $\|\mathbf{z}\|_q$
- Sparse coding: given D and $\mathbf{x}$ infer

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q$$

- Dictionary learning: given $\{x_i\}$ infer D s.t. $\|\mathsf{D}(:, j)\| \leq 1$ (why?)

$$\min_{\mathsf{D}} \sum_i \left(\min_{\mathbf{z}_i} \frac{1}{2} \|\mathbf{x}_i - \mathsf{D}\mathbf{z}_i\|^2 + \lambda \|\mathbf{z}_i\|_q \right)$$

- Usually not interpreted as probabilistic model
  - But: Bayesian sparse learning and automatic relevance determination

Sparse coding and dictionary learning

- Main objective

$$J(\mathsf{D}, \mathbf{z}) = \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q \qquad q \in [0, 1]$$

- Dictionary D
  - Columns of D are dictionary elements / atoms
  - Usually #rows(D) $\ll$ #columns(D)
- Sparse code $\mathbf{z}$: sparsity favored via $\|\mathbf{z}\|_q$
- Sparse coding: given D and $\mathbf{x}$ infer

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q$$

- Dictionary learning: given $\{x_i\}$ infer D s.t. $\|\mathsf{D}(:, j)\| \leq 1$ (why?)

$$\min_{\mathsf{D}} \sum_i \left(\min_{\mathbf{z}_i} \frac{1}{2} \|\mathbf{x}_i - \mathsf{D}\mathbf{z}_i\|^2 + \lambda \|\mathbf{z}_i\|_q \right)$$

- Usually not interpreted as probabilistic model
  - But: Bayesian sparse learning and automatic relevance determination

Sparse coding and dictionary learning

- Main objective

$$J(\mathsf{D}, \mathbf{z}) = \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q \qquad q \in [0, 1]$$

- Dictionary D
  - Columns of D are dictionary elements / atoms
  - Usually #rows(D) $\ll$ #columns(D)
- Sparse code $\mathbf{z}$: sparsity favored via $\|\mathbf{z}\|_q$
- Sparse coding: given D and $\mathbf{x}$ infer

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_q$$

- Dictionary learning: given $\{x_i\}$ infer D s.t. $\|\mathsf{D}(:, j)\| \leq 1$ (why?)

$$\min_{\mathsf{D}} \sum_i \left(\min_{\mathbf{z}_i} \frac{1}{2} \|\mathbf{x}_i - \mathsf{D}\mathbf{z}_i\|^2 + \lambda \|\mathbf{z}_i\|_q \right)$$

- Usually not interpreted as probabilistic model
  - But: Bayesian sparse learning and automatic relevance determination

Sparse coding and dictionary learning

- In many applications $q = 1$
- Convex but non-smooth sparse coding problem

$$\mathbf{z}(\mathbf{x}) = \arg\min_{z} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_1$$

- Iterative Shrinkage-Thresholding Algorithm (ISTA)
  - Proximal gradient descent
- Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)
  - Accelerated projected gradient descent

- Beck & Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”

Sparse coding and dictionary learning

- Sparse coding objective

$$\mathbf{z}(\mathbf{x}) = \arg\min_{z} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_1$$

- ISTA

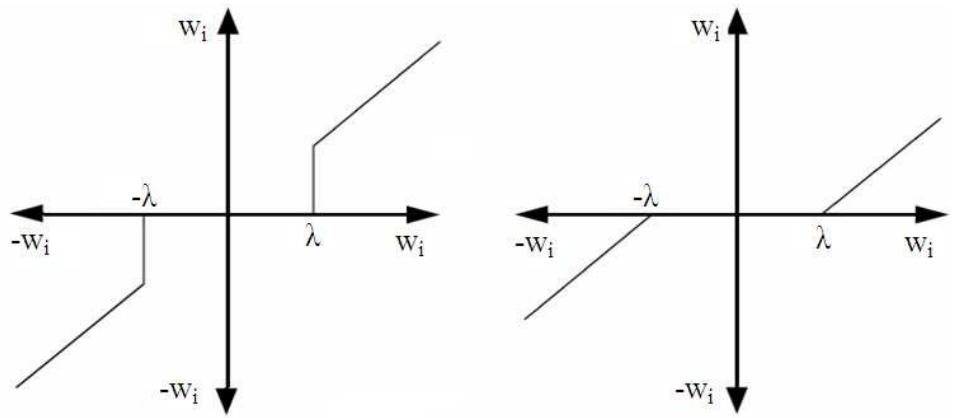
$$\tau \in (0, 1/\|\mathsf{D}^T\mathsf{D}\|_2), \mathbf{z}^{(0)} \leftarrow \mathbf{0}$$

$$\mathbf{z}^{(t+1/2)} \leftarrow \mathbf{z}^{(t)} - \tau \mathsf{D}^T(\mathsf{D}\mathbf{z}^{(t)} - \mathbf{x})$$

$$\mathbf{z}^{(t+1)} \leftarrow \arg\min_{\mathbf{z}} \lambda \|\mathbf{z}\|_1 + \frac{1}{2\tau} \left\|\mathbf{z} - \mathbf{z}^{(t+1/2)}\right\|^2$$

- Define

$$\mathcal{S}_\lambda(\mathbf{z}^0) := \arg\min_{\mathbf{z}} \lambda \|\mathbf{z}\|_1 + \frac{1}{2} \|\mathbf{z} - \mathbf{z}^0\|^2 = \begin{pmatrix} \vdots \\ [|z_j^0| - \lambda]_+ \operatorname{sgn}(z_j^0) \\ \vdots \end{pmatrix}$$



Sparse coding and dictionary learning

- Sparse coding objective

$$\mathbf{z}(\mathbf{x}) = \arg\min_{z} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \|\mathbf{z}\|_1$$

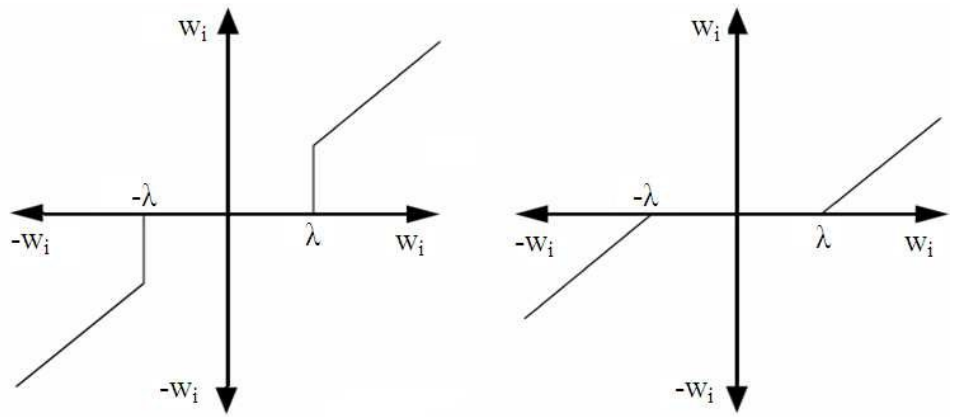
- ISTA in one line

$$\tau \in (0, 1/\|\mathsf{D}^T\mathsf{D}\|_2), \mathbf{z}^{(0)} \leftarrow \mathbf{0}$$

$$\mathbf{z}^{(t+1)} \leftarrow \mathcal{S}_{\tau\lambda}\left(\mathbf{z}^{(t)} - \tau\mathsf{D}^T(\mathsf{D}\mathbf{z}^{(t)} - \mathbf{x})^2\right)$$

- Define

$$\mathcal{S}_\lambda(\mathbf{z}^0) := \arg\min_{\mathbf{z}} \lambda \|\mathbf{z}\|_1 + \frac{1}{2} \|\mathbf{z} - \mathbf{z}^0\|^2 = \begin{pmatrix} \vdots \\ [|z_j^0| - \lambda]_+ \operatorname{sgn}(z_j^0) \\ \vdots \end{pmatrix}$$

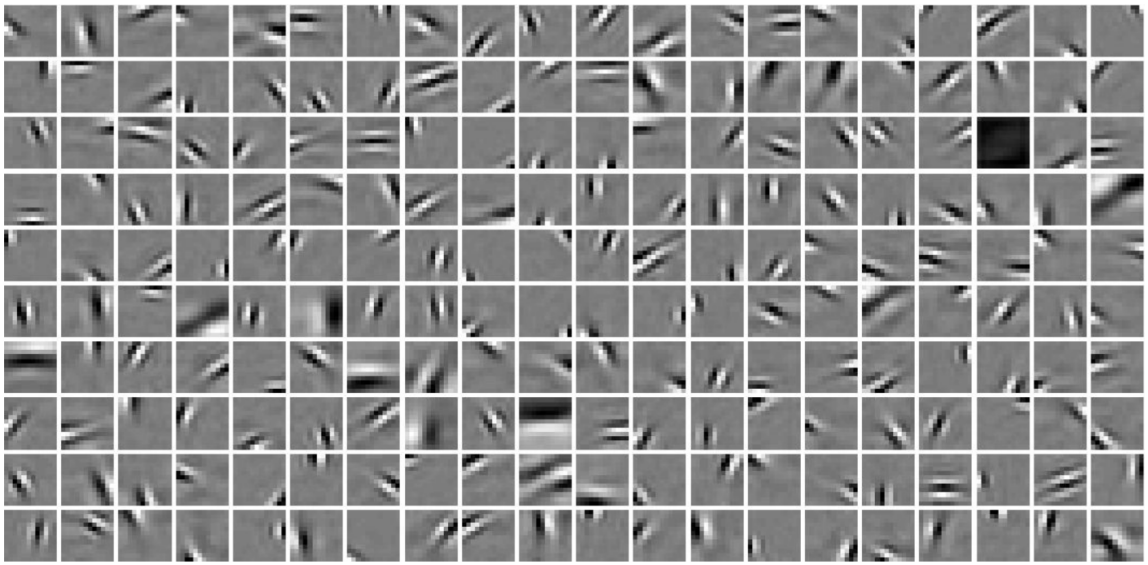


Sparse coding and dictionary learning

- Jointly learning sparse codes and dictionary

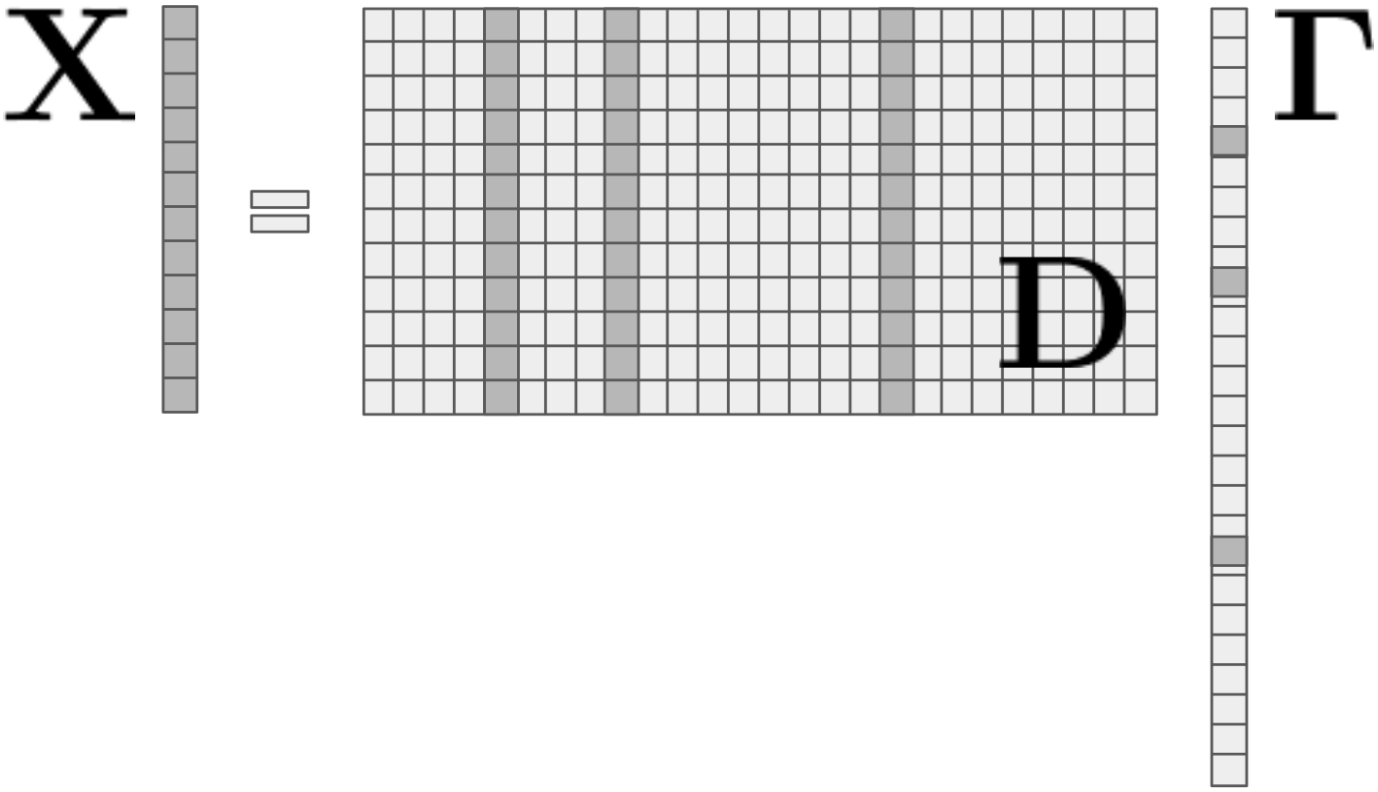
$$\min_{\mathsf{D}:\|\mathsf{D}(:,j)\|\leq 1} \min_{\{\mathbf{z}_i\}} \sum_i \left(\frac{1}{2} \|\mathbf{x}_i - \mathsf{D}\mathbf{z}_i\|^2 + \lambda \|\mathbf{z}_i\|_1 \right)$$

- Block-convex problem
- E.g. minimization by alternation
  - Find $\mathbf{z}_i$ for all i via FISTA
  - Update D via convex optimization (or column-by-column in closed form)



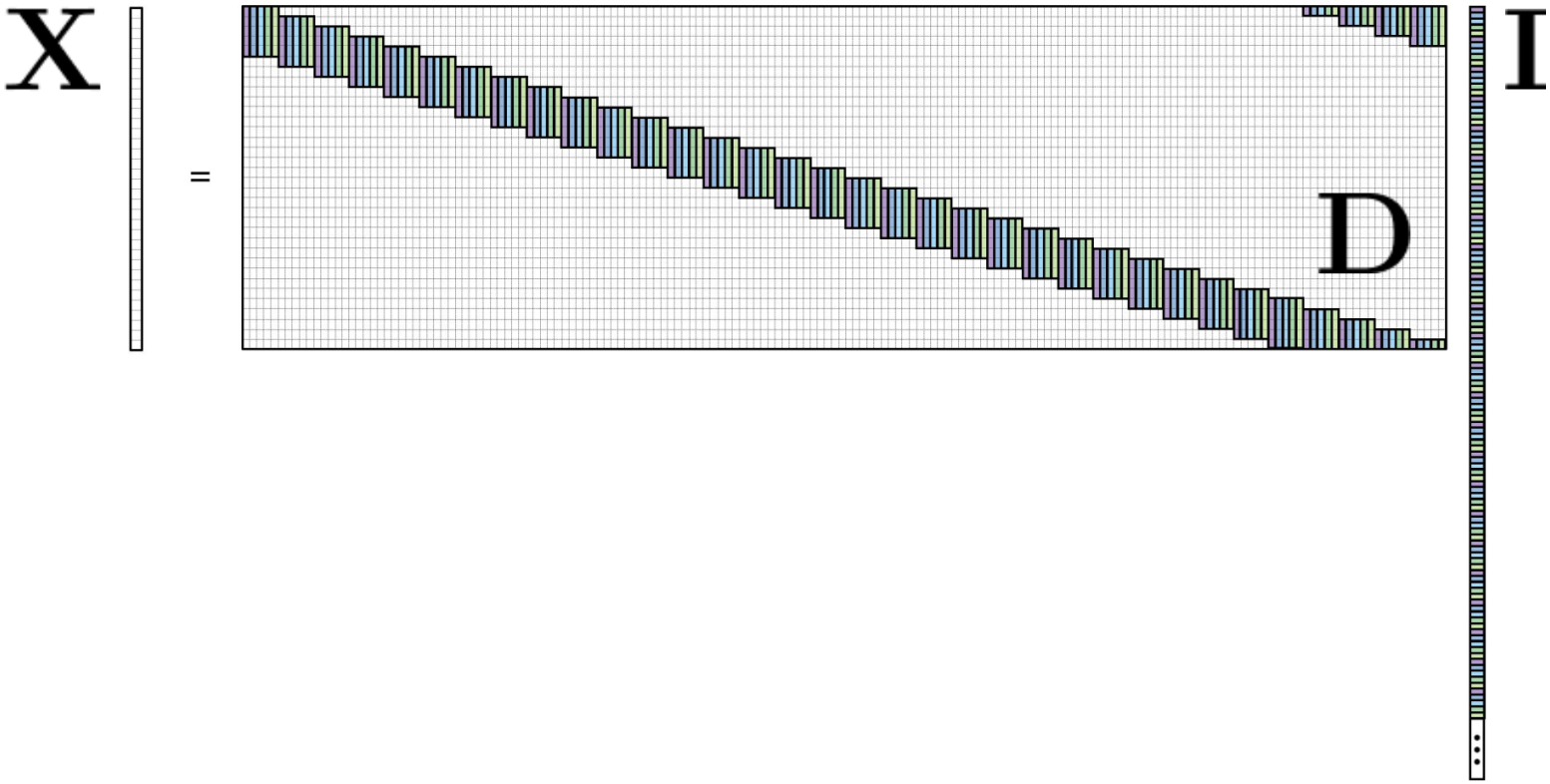
Convolutional sparse coding

- Going from patches to images
- Sparse coding

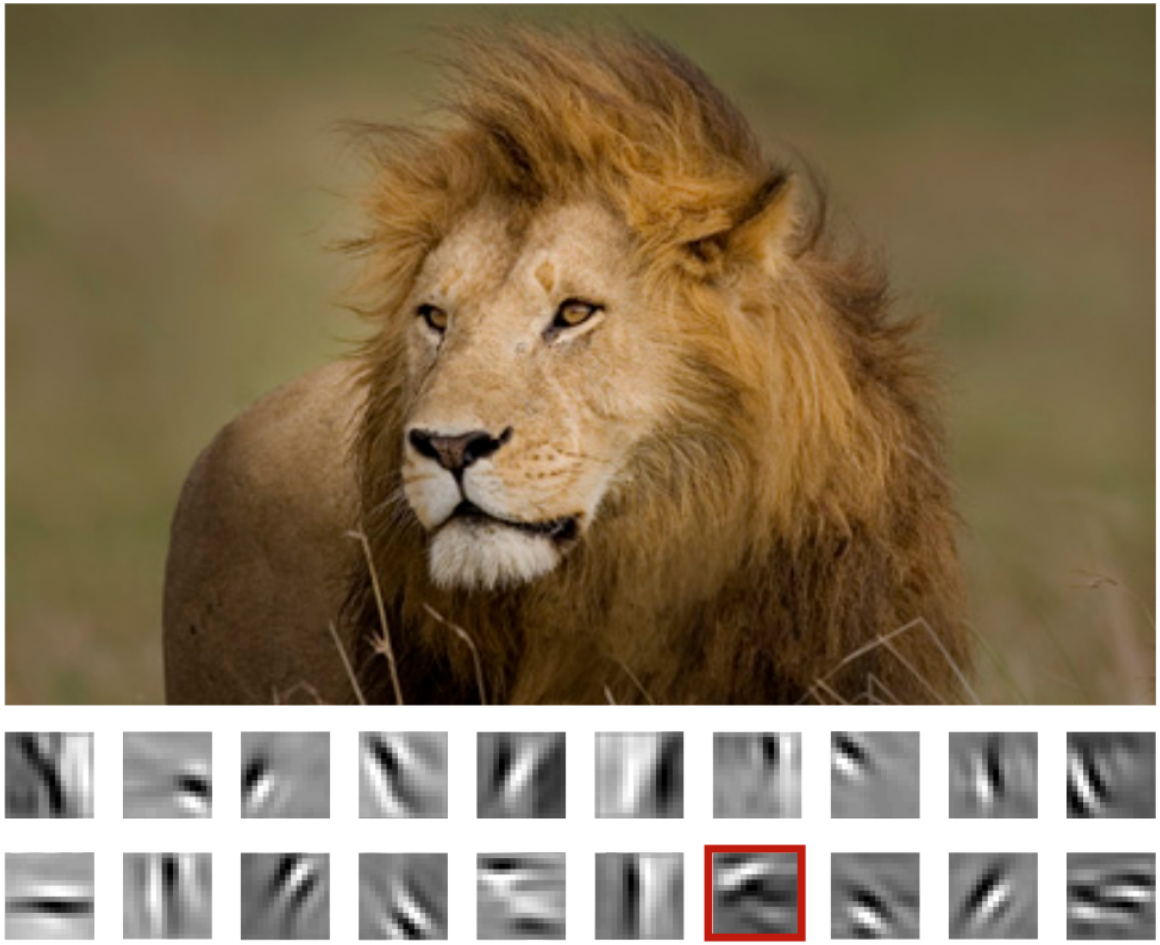


Convolutional sparse coding

- Going from patches to images
- Convolutional sparse coding



Convolutional sparse coding



- Bristow et al., “Fast Convolutional Sparse Coding”

Deep Sparse Coding

- Hierarchical (deep) sparse coding

$$\text{find } \{\mathbf{z}_k\}_{k=1}^{K} \text{ s.t. } \begin{aligned} &\|\mathbf{x} - \mathsf{D}_1\mathbf{z}_1\| \leq \varepsilon_1 && \|\mathbf{z}_1\|_1 \leq \mu_1 \\ &\|\mathbf{z}_1 - \mathsf{D}_1\mathbf{z}_2\| \leq \varepsilon_2 && \|\mathbf{z}_2\|_1 \leq \mu_2 \\ &\vdots && \vdots \\ &\|\mathbf{z}_{K-1} - \mathsf{D}_K\mathbf{z}_K\| \leq \varepsilon_K && \|\mathbf{z}_K\|_1 \leq \mu_K \end{aligned}$$

- Papyan et al., “Convolutional Neural Networks Analyzed via Convolutional Sparse Coding”

Deep Sparse Coding

- A layered thresholding algorithm

$$\mathbf{z}_1 \leftarrow \mathcal{S}_{\lambda_1}(\mathsf{D}_1^T \mathbf{x})$$
$$\mathbf{z}_2 \leftarrow \mathcal{S}_{\lambda_2}(\mathsf{D}_2^T \mathbf{z}_1)$$
$$\vdots$$
$$\mathbf{z}_K \leftarrow \mathcal{S}_{\lambda_K}(\mathsf{D}_K^T \mathbf{z}_{K-1})$$

- One step of ISTA (with $\mathbf{z}_k^{(0)} = 0$)
- Starts to look like a ReLU DNN
  - Non-negative sparse coding: ReLU DNN
  - Convolutional sparse coding: ReLU CNN

- Papyan et al., “Convolutional Neural Networks Analyzed via Convolutional Sparse Coding”

Deep Sparse Coding

- Non-negative sparse coding

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{D}\mathbf{z}\|^2 + \lambda \mathbf{z}$$

  subject to $\mathbf{z} \geq 0$

- Proximal gradient method

$$\mathbf{z}^{(0)} \leftarrow 0$$

$$\mathbf{z}^{(t+1)} \leftarrow \mathcal{S}^{+}_{\tau\lambda}\left(\mathbf{z}^{(t)} - \tau \mathsf{D}^T(\mathsf{D}\mathbf{z}^{(t)} - \mathbf{x})^2\right)$$

$$\mathcal{S}^{+}_{\lambda}(\mathbf{z}^0) := \arg\min_{\mathbf{z}\geq 0} \lambda \mathbf{z} + \frac{1}{2}\left\|\mathbf{z} - \mathbf{z}^0\right\|^2 = \begin{pmatrix} \vdots \\ [z_j^0 - \lambda]_+ \\ \vdots \end{pmatrix}$$

- Papyan et al., “Convolutional Neural Networks Analyzed via Convolutional Sparse Coding”

Deep Sparse Coding

- A layered thresholding algorithm for non-negative SC

$$\mathbf{z}_1 \leftarrow \mathcal{S}^+_{\lambda_1}(\mathsf{D}_1^T\mathbf{x}) = \left[\mathsf{D}_1^T\mathbf{x} - \lambda_1\right]_+$$
$$\mathbf{z}_2 \leftarrow \mathcal{S}^+_{\lambda_2}(\mathsf{D}_2^T\mathbf{z}_1) = \left[\mathsf{D}_2^T\mathbf{z}_1 - \lambda_2\right]_+$$
$$\vdots$$
$$\mathbf{z}_K \leftarrow \mathcal{S}^+_{\lambda_K}(\mathsf{D}_K^T\mathbf{z}_{K-1}) = \left[\mathsf{D}_K^T\mathbf{z}_{K-1} - \lambda_K\right]_+$$

- Looks like a ReLU DNN
- Convolutional sparse coding: ReLU CNN

Discussion

Is this analogy between deep sparse coding and DNNs too far fetched?

- Very crude approximation of ISTA
- Impact of batch normalization etc.

- Papyan et al., “Convolutional Neural Networks Analyzed via Convolutional Sparse Coding”

Deconvolutional networks

- Layerwise convolutional sparse coding

$$\mathbf{z}_l = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{z}_{l-1} - \mathsf{D}_l * \mathbf{z}\|^2 + \lambda_l \|\mathbf{z}_l\|_1$$

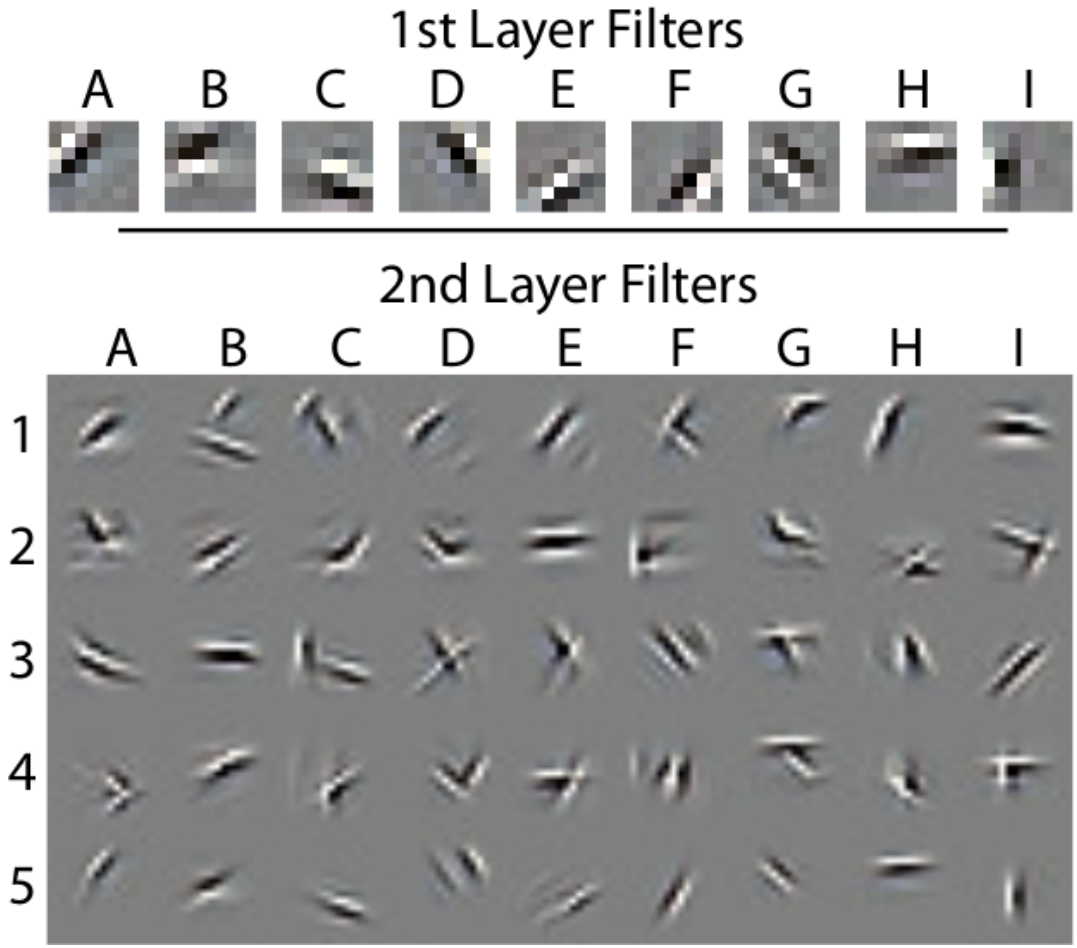
- Reconstruction operator by chaining, e.g.

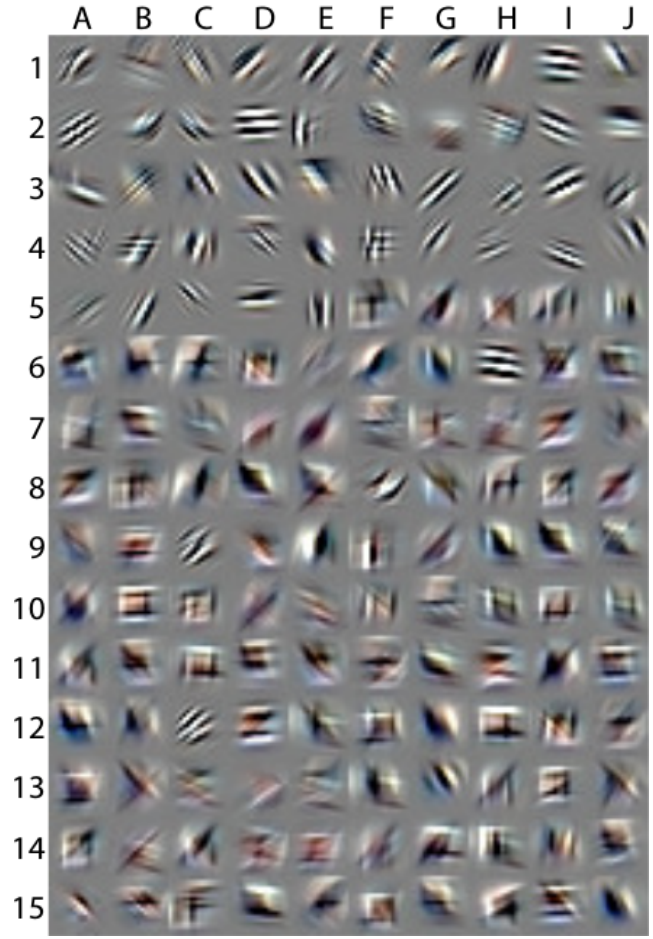
$$x' = \mathsf{D}_1 * (\mathsf{D}_2 * (\mathsf{D}_3 * \mathbf{z}_3))$$

- Layerwise training

- Zeiler et al., “Deconvolutional Networks”

Deconvolutional networks





- Zeiler et al., “Deconvolutional Networks”

Adaptive deconvolutional networks

- Add max-pooling / unpooling
  - Store pooling index to unpool
- Reconstruct input (instead of previous layer)

$$\mathbf{z}_l = \arg\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathsf{R}_l * \mathbf{z}\|^2 + \lambda_l \|\mathbf{z}_l\|_1$$

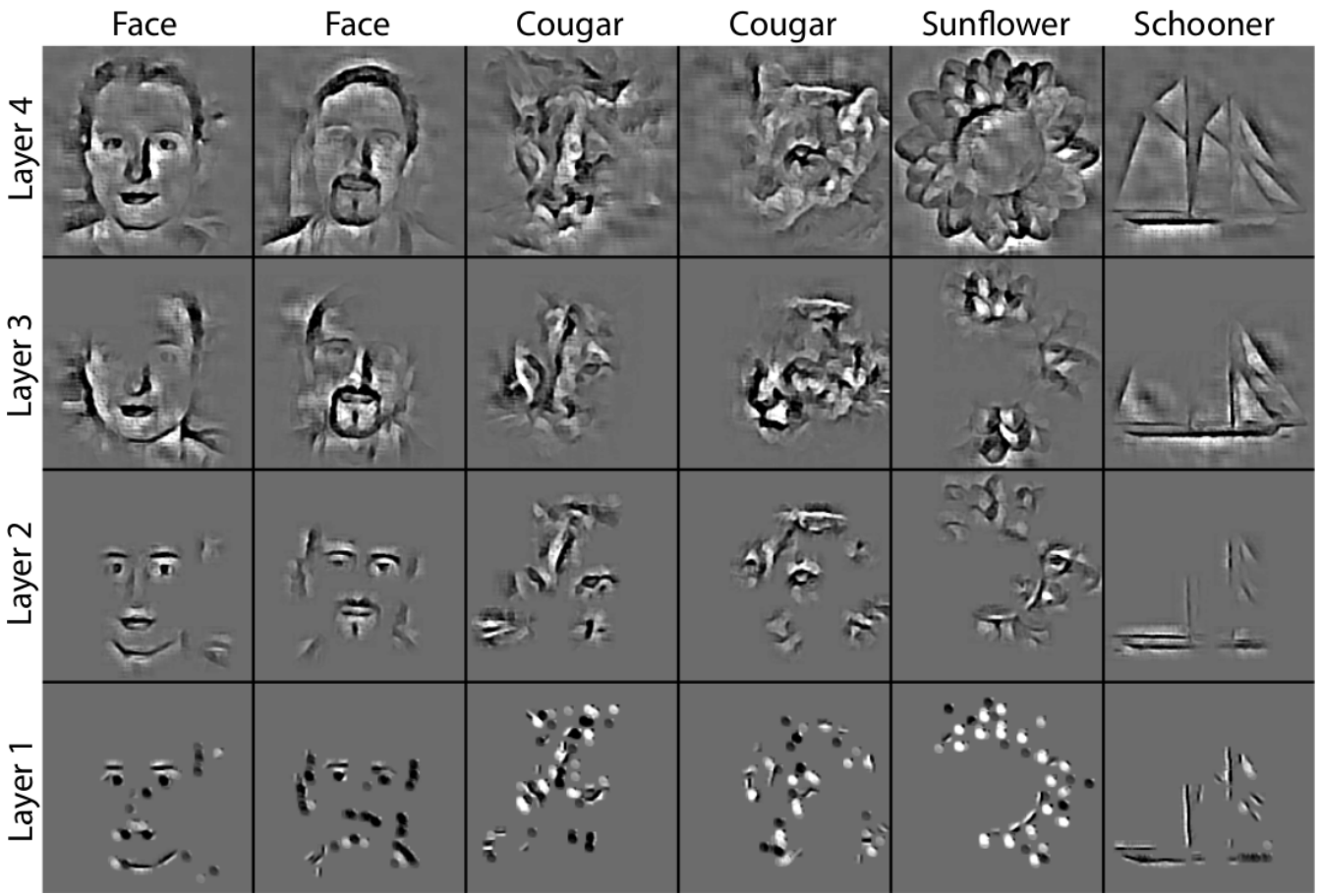
- Reconstruction operator by chaining, e.g.

$$x' = \mathsf{R}_3 \mathbf{z}_3 = \mathsf{D}_1 * \mathsf{U}_1(\mathsf{D}_2 * \mathsf{U}_2(\mathsf{D}_3 * \mathbf{z}_3))$$

- Layerwise training

- Zeiler et al., “Adaptive Deconvolutional Networks for Mid and High Level Feature Learning”

Adaptive deconvolutional networks



- Zeiler et al., “Adaptive Deconvolutional Networks for Mid and High Level Feature Learning”

Outline

Exercise 2: learning a two layer DEM on *your* image patches

Exercise

- Extract 50.000 random 28×28 natural image patches from a set of images
  - Holiday pictures, public domain pictures
- Convert patches to grayscale $\in [0, 1]^{28 \times 28}$
- Compute a (constrained) Gaussian representing this data
  - Computing the empirical mean
  - Using the method from exercise 1 to estimate a constrained precision matrix Λ
  - Use the method of your choice (SM, NCE, cNCE)
  - We will optionally whiten the patches using this estimate in the later steps

- Define $S(u) := \log(1 + e^u)$
- Attention: numerically robust way to compute S and $s = S'$

$$S(u) = \begin{cases} \log(1 + e^u) & \text{if } u \leq 0 \\ u + \log(1 + e^{-u}) & \text{if } u \geq 0 \end{cases} \qquad s(u) = \begin{cases} \frac{e^u}{1+e^u} & \text{if } u \leq 0 \\ \frac{1}{1+e^{-u}} & \text{if } u \geq 0 \end{cases}$$

Exercise 2: learning a two layer DEM on *your* image patches

Exercise

- Train a 2-layer DEM of the following form (using a method of your choice)

$$\log p_\theta(x) \doteq -\frac{1}{2\sigma^2}\|x\|^2 + b^T x + \sum_{k=1}^{K} S\left(w_k^T g_\theta(x) + c_k\right)$$

$$g_\theta(x) = s(Vx) \qquad \text{single layer sigmoid NN}$$

- Parameters: W, V, b, c e.g. $K = 64$, $V \in \mathbb{R}^{64\times 784}$
- Choose $\sigma \in \{1, 0.1\}$
- Run on whitened and non-whitened data
- Visualize filters v_j (rows in V)
- Is your model able to distinguish between
  - Hold-out natural patches
  - Samples generated by the fitted Gaussian ($\mathcal{N}(\mathbf{0}, \mathrm{I})$ in the whitened version)
  - MNIST digits
- Report mean and std. deviation of $\log p_\theta$
- Bonus: generate modes of p_θ by maximizing $\log p_\theta(x)$
  - Initial x: sample from fitted Gaussian