Variational Auto-Encoder & Normalizing Flows

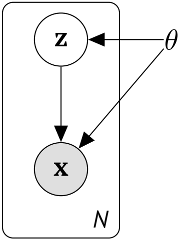
S. Lebedev, E. Tuzova

February 15, 2016

Motivation[1]

What are the underlying hidden factors in these two datasets?

[1] Slide credit: G. Hinton, CSC2515, <<Continuous Latent Variable Models>>.



x --- observed variables
z --- continuous latent variables
θ --- model parameters
$p(x, z|\theta) = p(x|z, \theta)p(z|\theta)$

- Goal: fast approximate posterior inference for the latent variables in the ``real-world'' scenario.
- Specifically when
  - non-conjugate distributions are involved,
  - so the evidence and posterior for latent variables are both intractable.
  - Mean-field VB is not applicable because the integrals required are intractable as well.

What are the options?

- MCMC
  - slow for large-scale problems,
  - diagnosing convergence is an issue.
- MAP
  - easy to overfit the data,
  - especially in the case of high-dimensional z.
- VB
  - mean-field cannot be applied directly,
  - but still a good idea,
  - maybe.

The plan

tl;dr reduce inference problem to stochastic optimization.

1. Approximate the posterior with a neural net $q(z|x, \phi)$, where ϕ --- variational parameters.
2. Lower bound the evidence using $q(z|x, \phi)$.
3. Construct an estimator of the ELBO which can be optimized jointly w.r.t. ϕ and θ.
4. Use stochastic gradient ascent to optimize the estimator.
5. Profit.

Having $q(z|x,\phi)$ we can deconstruct the evidence into

$$\begin{aligned}\log p(x|\theta) &= \log \int p(x,z|\theta)dz = \log \int q(z|x,\phi)\frac{p(x,z|\theta)}{q(z|x,\phi)}dz \\ &= \mathbb{D}_{KL}\left[q(z|x,\phi) \parallel p(z|x,\theta)\right] + \mathcal{L}(\theta,\phi;x) \\ &\geq \mathcal{L}(\theta,\phi;x)\end{aligned}$$

where the lower bound is given by

$$\begin{aligned}\mathcal{L}(\theta,\phi;x) &= \mathbb{E}_{q(z|x,\phi)}\left[\log p(x,z|\theta) - \log q(z|x,\phi)\right] \\ &= \mathbb{E}_{q(z|x,\phi)}\left[\log p(x|z,\theta)\right] - \mathbb{D}_{KL}\left[q(z|x,\phi) \parallel p(z|\theta)\right]\end{aligned}$$

- Want to optimize the lower bound w.r.t both θ and ϕ.
- Just-do-it approach:

$$\begin{aligned}
\nabla_\theta \mathcal{L}(\theta, \phi; x) &= \nabla_\theta \mathbb{E}_{q(z|x,\phi)} [\log p(x, z|\theta) - \log q(z|x, \phi)] \\
&= \mathbb{E}_{q(z|x,\phi)} [\nabla_\theta \log p(x, z|\theta)] \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\theta \log p(x, z^{(s)}|\theta) \\
\nabla_\phi \mathcal{L}(\theta, \phi; x) &= \nabla_\phi \mathbb{E}_{q(z|x,\phi)} [\log p(x, z|\theta) - \log q(z|x, \phi)]
\end{aligned}$$

- How to deal with gradients of the form $\nabla_\phi \mathbb{E}_{q(z|\phi)} [f(z)]$?

Naïve MCMC estimator of $\nabla_\phi \mathbb{E}_{q(z|\phi)}[f(z)]$

$$
\begin{aligned}
\nabla_\phi \mathbb{E}_{q(z|\phi)}[f(z)] &= \nabla_\phi \int q(z|\phi) f(z) dz = \int f(z) \nabla_\phi q(z|\phi) dz \\
&= \int f(z) q(z|\phi) \nabla_\phi \log q(z|\phi) dz
\end{aligned}
$$

where the last line is due to the log derivative trick [2]

$$
\nabla_\phi \log q(z|x,\phi) = \frac{\nabla_\phi q(z|x,\phi)}{q(z|x,\phi)}
$$

Proceeding further we obtain

$$
\begin{aligned}
\nabla_\phi \mathbb{E}_{q(z|\phi)}[f(z)] &= \mathbb{E}_{q(z|x,\phi)}[f(z) \nabla_\phi \log q(z|\phi)] \\
&\approx \frac{1}{S} \sum_{s=1}^{S} f(z^{(s)}) \nabla_\phi \log q(z^{(s)}|\phi) \rightarrow :(
\end{aligned}
$$

[2] http://blog.shakirm.com/2015/11/machine-learning-trick-of-the-day-5-log-derivative-trick

Reparametrization trick[3]

- Introduce an auxilary noise variable ϵ independent of ϕ.
- Express z as a determinisitic transformation of ϵ differentiable w.r.t. ϕ

$$z = g(\epsilon, x; \phi) \qquad \epsilon \sim p(\epsilon)$$

- Example: let $q(z|\phi) = \mathcal{N}(z|\mu, \sigma^2)$ and $\phi = (\mu, \sigma^2)$ then

$$\begin{aligned} \nabla_\phi \mathbb{E}_{q(z|\phi)}\left[f(z)\right] &= \mathbb{E}_{\mathcal{N}(\epsilon|0,1)}\left[\nabla_\phi f(\mu + \sigma\epsilon)\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi f(\mu + \sigma\epsilon^{(s)}) \end{aligned}$$

where $z = \mu + \sigma\epsilon$ and $\epsilon \sim \mathcal{N}(0, 1)$.

[3] http://blog.shakirm.com/2015/10/machine-learning-trick-of-the-day-4-reparameterisation-tricks

- In general

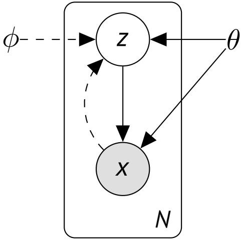
$$\begin{aligned}\mathcal{L}(\theta,\phi;x) &= \mathbb{E}_{q(z|x,\phi)}\left[\log p(x,z|\theta) - \log q(z|x,\phi)\right]\\ &\approx \frac{1}{S}\sum_{s=1}^{S}\log p(x,z^{(s)}|\theta) - \log q(z^{(s)}|x,\phi)\\ &\triangleq \widetilde{\mathcal{L}}(\theta,\phi;x,z)\end{aligned}$$

  where $z^{(s)} = g(\epsilon^{(s)},x;\phi)$ and $\epsilon^{(s)} \sim p(\epsilon)$.
- If $\mathbb{D}_{KL}\left[q(z|x,\phi)\parallel p(z|\theta)\right]$ is tractable

$$\begin{aligned}\mathcal{L}(\theta,\phi;x) &= \mathbb{E}_{q(z|x,\phi)}\left[\log p(x|z,\theta)\right] - \mathbb{D}_{KL}\left[q(z|x,\phi)\parallel p(z|\theta)\right]\\ &\approx \frac{1}{S}\sum_{s=1}^{S}\log p(x|z^{(s)},\theta) - \mathbb{D}_{KL}\left[q(z|x,\phi)\parallel p(z|\theta)\right]\\ &\triangleq \widetilde{\mathcal{L}}(\theta,\phi;x,z)\end{aligned}$$

AEVB algorithm [?]

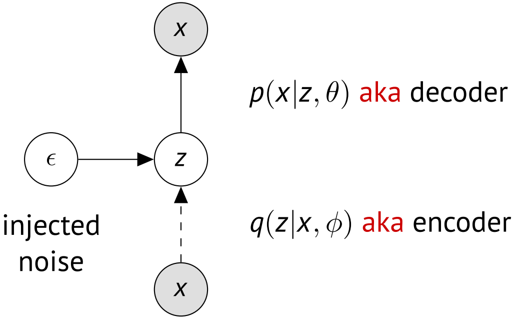
$\alpha \leftarrow$ set learning rate
$\theta, \phi \leftarrow$ initialize parameters
repeat
    $x \leftarrow$ random datapoint or minibatch
    $\epsilon \leftarrow$ random samples from $p(\epsilon)$
    $z \leftarrow g(\epsilon, x; \phi)$
    $g_\theta, g_\phi \leftarrow \nabla_{\phi,\theta} \widetilde{\mathcal{L}}(\theta, \phi; x, z)$
    $\theta \leftarrow \theta + \alpha g_\theta$
    $\phi \leftarrow \phi + \alpha g_\phi$
until convergence
return θ, ϕ



$$p(z|\theta) = \mathcal{N}(z|0, \mathbf{I})$$
$$p(x|z,\theta) = \mathcal{N}(x|\mu(z), \sigma^2(z)\mathbf{I})$$
$$q(z|x,\phi) = \mathcal{N}(z|M(x), S^2(x)\mathbf{I})$$

- The parameters $M(x)$ and $S^2(x)$ are computed by a neural net, which assigns each value of x a distribution over z.
- The parameters $\mu(z)$ and $\sigma^2(z)$ are computed by a different neural net, mapping z to a distribution over x.

VAE illustrated

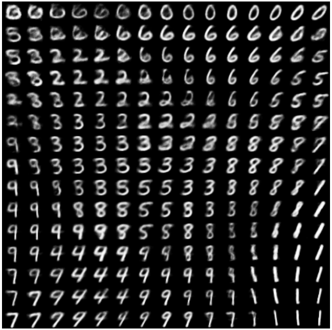


$$\mathcal{L}(\theta,\phi;x) = \underbrace{\mathbb{E}_{q(z|x,\phi)}\left[\log p(x|z,\theta)\right]}_{\text{negative reconstruction error}} - \underbrace{\mathbb{D}_{KL}\left[q(z|x,\phi) \parallel p(z|\theta)\right]}_{\text{regularizer}}$$

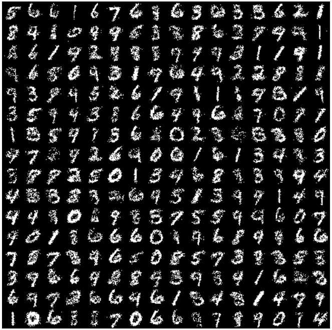
Experiments: Frey faces


(a) Learned data manifold


(b) Random samples



(a) Learned data manifold



(b) Random samples

What is wrong with VAE?

We want to specify a complex joint distribution over z.

z --- random variable with distribution $q(z)$
f --- invertible parametric function

Transformation of random variables: $\tilde{z} = f(z),\ f^{-1}(\tilde{z}) = z$

$$q(\tilde{z}) = q(f^{-1}(\tilde{z})) \left| \det \frac{\partial f^{-1}(\tilde{z})}{\partial \tilde{z}} \right| = q(z) \left| \det \frac{\partial f(z)}{\partial z} \right|^{-1}$$

Chaining together a sequence: $z_K = f_K(f_{K-1}(\cdots f_2(f_1(z_0)))$

$$\log q_K(z_K) = \log q_0(z_0) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial z_k} \right|$$

Law of the unconscious statistician:

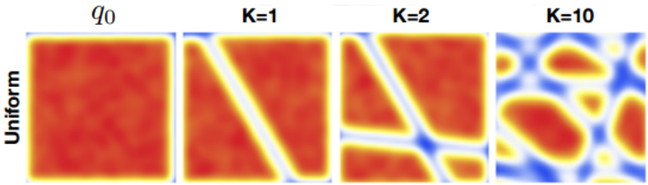
$$\mathbb{E}_{q_K}\left[g(z_K)\right] = \mathbb{E}_{q_0}\left[g(f_K(f_{K-1}(\cdots f_2(f_1(z_0))))\right]$$

Family of transformations: $f(z) = z + uh\left(w^T z + b\right)$

$$\left|\det \frac{\partial f(z)}{\partial z}\right| = \left|1 + u^T \psi(z)\right| \quad \text{where} \quad \psi(z) = h'(w^T z + b)w$$

$$\log q_K(z_K) = \log q_0(z_0) - \sum_{k=1}^{K} \log \left|1 + u^T \psi(z)\right|$$

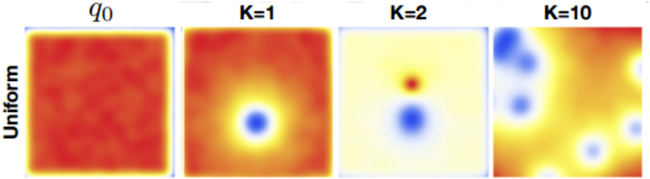
Chaining transformations gives us a rich family of densities.



Family of transformations: $f(z) = z + \beta h(\alpha, r)(z - z_0)$
$$r = |z - z_0|, \quad h(\alpha, r) = \frac{1}{\alpha + r}$$

$$\left|\det \frac{\partial f}{\partial z}\right| = [1 + \beta h(\alpha, r)]^{(d-1)}[1 + \beta h(\alpha, r) + h'(\alpha, r)r]$$

Chaining transformations gives us a rich family of densities.



Representative power of normalizing flows

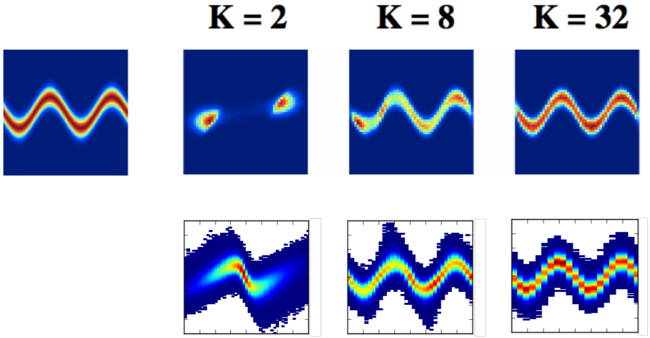
- Choose a non-trivial density $p(z) \propto \exp[-U(z)]$.
- Example:

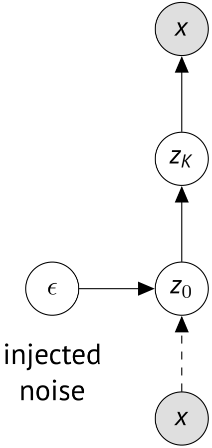
$$U(z) = \frac{1}{2}\left(\frac{z_2 - w_1(z)}{0.4}\right) \qquad w_1(z) = \sin\frac{2\pi z_1}{4}$$

- Approximate the density with a flow by optimizing

$$\begin{aligned}
\mathbb{D}_{KL}\left[q(z_K) \parallel p(z)\right] &= \int q(z_K) \log \frac{q(z_K)}{p(z)} dz_K \\
&= \mathbb{E}_{q(z_K)}\left[\log q(z_K) - (-U(z) + \text{const}(z_K))\right] \\
&\approx \frac{1}{S}\sum_{s=1}^{S} (\log q(z_K) + U(z)) + \text{const}(z_K)
\end{aligned}$$

Representative power of planar flows





$p(x|z_K, \theta)$ aka decoder

normalizing flow

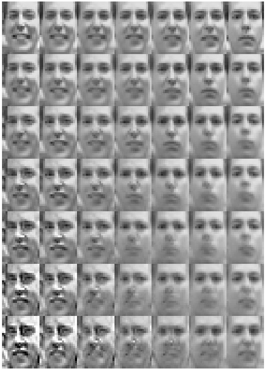
$q(z_0|x, \phi)$ aka encoder

ELBO with planar normalizing flow

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, x) &= \mathbb{E}_{q(z|x,\phi)} \left[\log p(x, z|\theta) - \log q(z|x, \phi)\right] \\
&= \mathbb{E}_{q_K(z_K|\theta)} \left[\log p(x, z_K|\theta) - \log q(z_K|\phi)\right] \\
&= \mathbb{E}_{q_0(z_0|x,\phi)} \Bigg[\log p(x, z_K|\theta) - \log q_0(z_0|x, \phi) \\
&\qquad + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial z_k} \right| \Bigg]
\end{aligned}
$$

$\alpha \leftarrow$ set learning rate
$\theta, \phi \leftarrow$ initialize parameters
repeat
    $x \leftarrow$ random datapoint or minibatch
    $\epsilon \leftarrow$ random samples from $p(\epsilon)$
    $z_0 \leftarrow g(\epsilon, x; \phi)$
    $z_K \leftarrow f_K(f_{K-1}(\cdots f_1(z_0)))$
    $g_\theta, g_\phi \leftarrow \nabla_{\phi,\theta} \widetilde{\mathcal{L}}(\theta, \phi; x, z_K)$
    $\theta \leftarrow \theta + \alpha g_\theta$
    $\phi \leftarrow \phi + \alpha g_\phi$
until convergence
return θ, ϕ

Experiments: Frey faces ($K = 2$)



(a) Learned data manifold

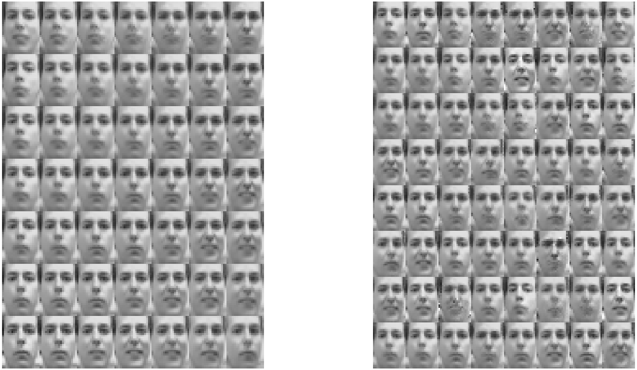
(b) Random samples

Experiments: Frey faces ($K = 8$)

(a) Learned data manifold

(b) Random samples

Experiments: Frey faces ($K = 16$)



(a) Learned data manifold

(b) Random samples

Experiments: Frey faces

Model	**ELBO**
VAE	519.72
NF ($K = 2$)	331.27
NF ($K = 8$)	410.03
NF ($K = 16$)	415.49

Experiments: MNIST ($K = 2$)

(a) Learned data manifold

(b) Random samples

- Investigate the effect of latent variable prior $p(z)$ and approximate posterior $q(z_K|x, \phi)$ on model performance.
- Try more complex prior distributions for the case when domain-specific knowledge is available, e.g. the data is multimodal.
- Apply normalizing flows to the problem of semi-supervised learning with generative models.