Introduction to Variational Autoencoders

CS 598: Deep Generative and Dynamical Models

Instructor: Arindam Banerjee

August 31, 2021

Latent Variable Models, Redux

- Joint distribution of a latent variable model (LVM)

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

  - x denotes the observed variable
  - z denotes the latent variable
  - θ denotes the parameters
- Problems of interest
  - Compute marginal or conditional distributions

$$p_\theta(x) = \int_z p_\theta(x, z)dz \qquad p_\theta(z|x) = \frac{p_\theta(x, z)}{p_\theta(x)}$$

  - Estimate θ by optimizing a function of $p_\theta(x)$
- Problems need to compute high-d integrals

Kingma and Welling, ICLR 2014

We are interested in, and propose a solution to, three related problems in the above scenario:

1. Efficient approximate ML or MAP estimation for the parameters $\boldsymbol{\theta}$. The parameters can be of interest themselves, e.g. if we are analyzing some natural process. They also allow us to mimic the hidden random process and generate artificial data that resembles the real data.
2. Efficient approximate posterior inference of the latent variable $\mathbf{z}$ given an observed value $\mathbf{x}$ for a choice of parameters $\boldsymbol{\theta}$. This is useful for coding or data representation tasks.
3. Efficient approximate marginal inference of the variable $\mathbf{x}$. This allows us to perform all kinds of inference tasks where a prior over $\mathbf{x}$ is required. Common applications in computer vision include image denoising, inpainting and super-resolution.

Variational Autoencoders (VAE): Two Ideas

- Construct a distribution $q_\phi(\mathrm{z}|\mathrm{x})$ with parameters ϕ
- Choose family q and parameters ϕ to approximate true posterior

$$q_\phi(\mathrm{z}|\mathrm{x}) \approx p_\theta(\mathrm{z}|\mathrm{x})$$

- Two Ideas
  - Use a deep network as inference model (encoder) $q_\phi(z|x)$
  - Compute gradients w.r.t. ϕ by reparametrization
- Also, deep generative model $p_\theta(\mathrm{x}|\mathrm{z})$ (decoder)

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x)\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{p_\theta(z|x)}\right)\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\right]}_{\mathcal{L}_{\theta,\phi}(x)\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right]}_{D_{KL}(q_\phi(z|x)\|p_\theta(z|x))}
\end{aligned}
$$

Maximize the ELBO, lower bound to $\log p_\theta(x)$

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)]$$

Forms of the ELBO

- Entropy (regularized) form, $H(q) = \mathbb{E}_q[-\log q]$

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

- Regularized (negative) reconstruction error form

$$\text{(max)} \quad \mathcal{L}_{\theta,\phi}(x) = -D_{KL}(q_\phi(z|x)\|p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$$

$$\text{(min)} \quad -\mathcal{L}_{\theta,\phi}(x) = \underbrace{D_{KL}(q_\phi(z|x)\|p_\theta(z))}_{\text{Regularization}} + \underbrace{\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)]}_{\text{reconstruction error}}$$

  - Regularization encourages $q_\phi(z|x)$ to stay close to the prior $p_\theta(z)$
  - Reconstruction error: Encoding $q_\phi : x \mapsto z$, decoding $p_\theta : z \mapsto x$
  - Low error: High $q_\phi(z|x)$ regions should map to high $p_\theta(x|z)$ regions
    - Probabilistic encoding $q_\phi(z|x)$ and decoding $p_\theta(x|z)$

Inference Model

- Inference is done based on a dataset $\{x_i, i = 1, \ldots, n\}$
- Mean field VI assumes a tractable inference model

$$q_\phi(\mathrm{z}|\mathrm{x}) = \prod_{i=1}^{n} q_{\phi_i}(z_i|x_i)$$

  - Naive mean field, fully factorized distribution over $\{z_{ij}\}$
  - More generally, finite tree-width dependence
- Optimize over the *free* variational parameters $\{\phi_i, i = 1, \ldots, n\}$
  - Optimizing each ϕ_i slow for large datasets
- Inference model (deep network) $q_\phi(\mathrm{z}_i|\mathrm{x}_i)$
  - Global model with parameter ϕ, optimized using all data
  - Generalized (non-linear) model q_ϕ, e.g., parameters as $g_\phi(\mathrm{x}_i)$
  - No need for separate parameters ϕ_i, more efficient

Optimizing the ELBO w.r.t. θ

- Monte Carlo (MC) samples $z_{i,\ell} \sim q_\phi(\mathsf{z}|\mathsf{x}_i), \ell = 1, \ldots, L$
- Gradient of ELBO w.r.t. θ can be approximated using the samples

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\theta,\phi}(\mathsf{x}_i) &= \nabla_\theta \mathbb{E}_{q_\phi(\mathsf{z}|\mathsf{x}_i)}[\log p_\theta(\mathsf{x}_i, \mathsf{z}) - \log q_\phi(\mathsf{z}|\mathsf{x}_i)] \\
&= \mathbb{E}_{q_\phi(\mathsf{z}|\mathsf{x}_i)}[\nabla_\theta(\log p_\theta(\mathsf{x}_i, \mathsf{z}) - \log q_\phi(\mathsf{z}|\mathsf{x}_i))] \\
&\approx \frac{1}{L}\sum_{\ell=1}^{L} \nabla_\theta(\log p_\theta(\mathsf{x}_i, \mathsf{z}_{i,\ell}) - \log q_\phi(\mathsf{z}_{i,\ell}|\mathsf{x}_i)) \\
&= \frac{1}{L}\sum_{\ell=1}^{L} \nabla_\theta(\log p_\theta(\mathsf{x}_i, \mathsf{z}_{i,\ell})
\end{aligned}$$

- Implement using a mini-batch of saples $\{\mathsf{x}_i\}$
- In practice, usually $L = 1$

Optimizing the ELBO w.r.t. ϕ

- Gradient of ELBO w.r.t. ϕ is tricky

$$\nabla_\phi \mathcal{L}_{\theta,\phi}(x_i) = \nabla_\phi \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i, z) - \log q_\phi(z|x_i)]$$
$$\neq \mathbb{E}_{q_\phi(z|x_i)}[\nabla_\phi(\log p_\theta(x_i, z) - \log q_\phi(z|x_i))]$$

  - Expectation is over $q_\phi(z|x_i)$ which depends on ϕ
  - Gradient of second (entropy) term often tractable

- Approach based on MC samples for first term

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x_i)}[f(z, x_i)] = \mathbb{E}_{q_\phi(z|x_i)}[f(z, x_i)\nabla_\phi \log q_\phi(z|x_i)]$$
$$\approx \frac{1}{L}\sum_{\ell=1}^{L} f(z_{i,\ell}, x_i)\nabla_\phi \log q_\phi(z_{i,\ell}|x_i)$$

  - Called REINFORCE, likelihood ratio estimator, ...
  - Has high variance, need methods for variance control

Reparameterization

- Random variables can be reparameterized, under some conditions

$$z \sim q_\phi(z|x) \quad \equiv \quad z = g_\phi(\epsilon, x) \ , \quad \epsilon \sim p(\epsilon)$$

- Example:
  $z \sim \mathcal{N}(\mu(x), \sigma^2(x)) \quad \equiv \quad z = \mu(x) + \sigma(x)\epsilon, \epsilon \sim \mathcal{N}(0, 1)$
- Decoupling parameters and source of randomness
- MC estimates become possible, with $\epsilon_{i,\ell} \sim p(\epsilon)$

$$\mathbb{E}_{q_\phi(z|x_i)}[f(z, x_i)] = \mathbb{E}_{p(\epsilon)}[f(g_\phi(\epsilon, x_i), x_i)] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(g_\phi(\epsilon_{i,\ell}, x_i), x_i)$$

- Gradient w.r.t. ϕ now becomes possible

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x_i)}[f(z, x_i)] \approx \frac{1}{L}\sum_{\ell=1}^{L} \nabla_\phi f(g_\phi(\epsilon_{i,\ell}, x_i), x_i)$$

Reparameterization

- MC sample based estimate of the ELBO

$$\mathcal{L}^{A}_{\theta,\phi}(\mathrm{x}) = \frac{1}{L}\sum_{\ell=1}^{L} \log p_\theta(\mathrm{x}_i, \mathrm{z}_{i,\ell}) - \log q_\phi(\mathrm{z}_{i,\ell}|\mathrm{x}_i)$$

$$\text{where} \quad \mathrm{z}_{i,\ell} = g_\phi(\epsilon_{i,\ell}, \mathrm{x}_i) \ , \quad \epsilon_{i,\ell} \sim p(\epsilon)$$

- KL-divergence term can be analytically computed, in some settings

$$\mathcal{L}^{B}_{\theta,\phi}(\mathrm{x}) = -D_{KL}(q_\phi(\mathrm{z}|\mathrm{x}_i)\|p_\theta(\mathrm{z})) + \frac{1}{L}\sum_{\ell=1}^{L} \log p_\theta(\mathrm{x}_i|\mathrm{z}_{i,\ell})$$

$$\text{where} \quad \mathrm{z}_{i,\ell} = g_\phi(\epsilon_{i,\ell}, \mathrm{x}_i) \ , \quad \epsilon_{i,\ell} \sim p(\epsilon)$$

Auto-Encoding VB Algorithm

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

Approaches to Reparameterization

Random variables can be reparameterized, under some conditions

$$z \sim q_\phi(z|x) \quad \equiv \quad z = g_\phi(\epsilon, x) \ , \quad \epsilon \sim p(\epsilon)$$

- Tractable inverse CDF
  - Sample $\epsilon \sim \mathcal{U}[0,1]$, sample $z = Q_\phi^{-1}(\epsilon)$ for CDF Q_ϕ
  - $g_\phi(\epsilon, x)$ is inverse CDF of $q_\phi(z|x)$, $z = g_\phi(\epsilon, x)$
  - Examples: Exponential, Weibull, Gumbel, Erlang, etc.
- Location-scale family, similar to Gaussians
  - Use $g(\cdot) = \text{location} + \text{scale} \cdot \epsilon$
  - Examples: Elliptical, Student's t, Uniform, Gaussian, etc.
- Composition or transformations of base variables
  - Log-normal, Gamma, Dirichlet, Beta, Chi-squared, etc.

Variational Autoencoder

- Generative model: Deep network for $p_\theta(\mathrm{x}|\mathrm{z})$
  - Generalized (nonlinear) model for Gaussian or Bernoulli parameters
  - Prior $p_\theta(\mathrm{z}) = \mathcal{N}(\mathrm{z}; 0, \mathbb{I})$, parameter free
  - True posterior $p_\theta(\mathrm{z}|\mathrm{x}) = \frac{p_\theta(\mathrm{z})p_\theta(\mathrm{x}|\mathrm{z})}{p_\theta(\mathrm{x})}$ is intractable
- Inference model: Deep network for $q_\phi(\mathrm{z}|\mathrm{x})$

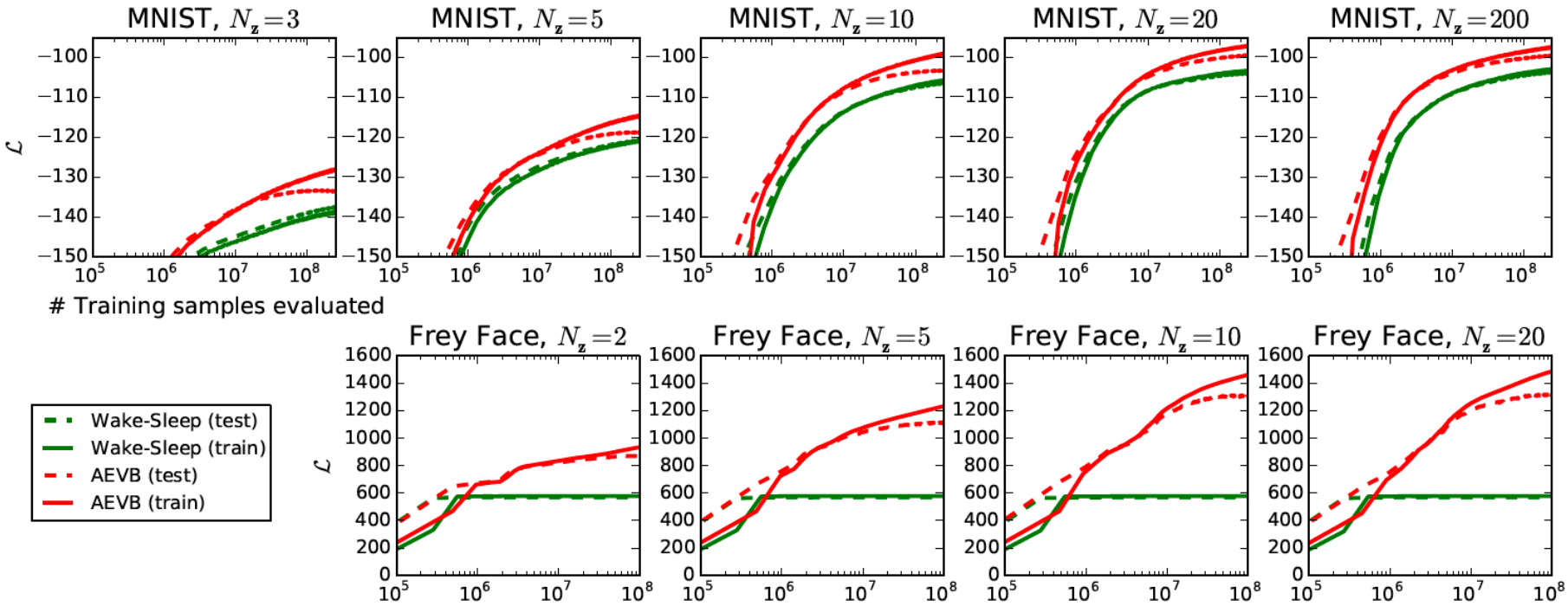
$$q_\phi(\mathrm{z}|\mathrm{x}_i) = \mathcal{N}(\mathrm{z}; \mu_\phi(\mathrm{x}_i), \sigma^2_\phi(\mathrm{x}_i)\mathbb{I})$$

  - Denote $\mu_i = \mu_\phi(\mathrm{x}_i) \in \mathbb{R}^p$, $\sigma^2_i = \sigma^2_\phi(\mathrm{x}_i) \in \mathbb{R}^p$
  - Reparameterization can be done with $\epsilon_{i,l} \sim \mathcal{N}(0, \mathbb{I})$
- MC sample based ELBO

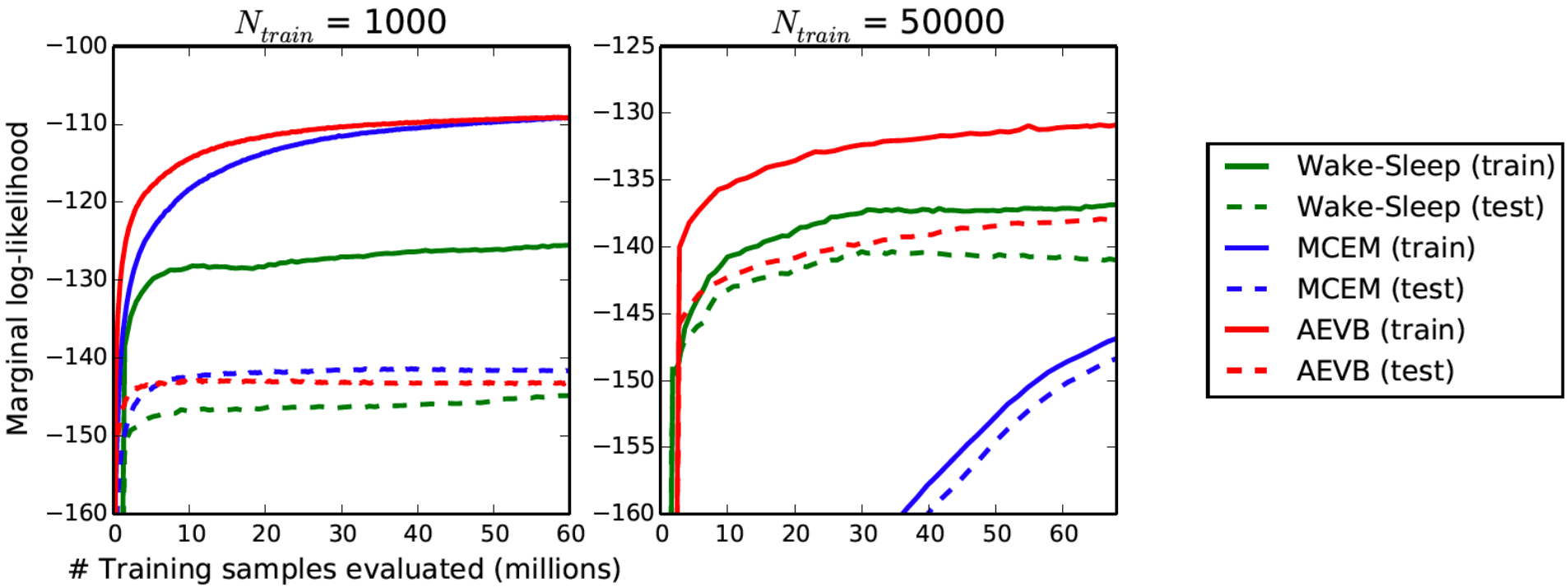
$$\mathcal{L}_{\theta,\phi}(\mathrm{x}) = \frac{1}{2}\sum_{j=1}^{p}\left(1 + \log(\sigma^2_{i,j}) - \mu^2_{i,j} - \sigma^2_{i,j}\right) + \frac{1}{L}\sum_{\ell=1}^{L}\log p_\theta(\mathrm{x}_i|\mathrm{z}_{i,\ell})$$

where $\quad \mathrm{z}_{i,\ell} = \mu_i + \sigma_i \odot \epsilon_{i,\ell}\ , \quad \epsilon_{i,\ell} \sim \mathcal{N}(0, \mathbb{I})$

Results: Lower Bound Optimization

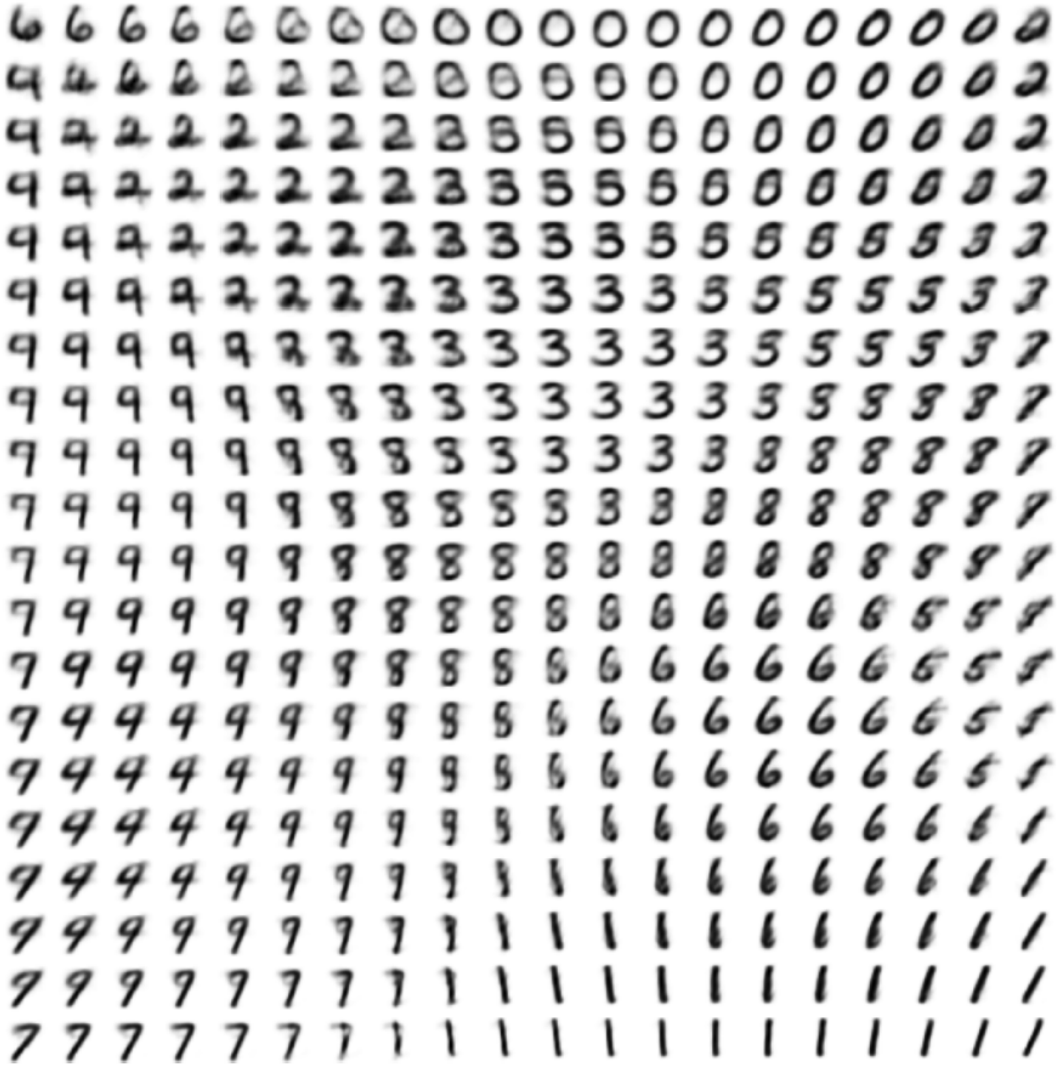


Results: Marginal Likelihood, Sample Efficiency



Results: Latent Space Embedding, Visualization

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Deep Latent Gaussian Models

- Generative model with layers of latent variables
  - D layers of latent variables z_d
  - Non-linear transformations and Gaussian convolutions
  - Last layer generates observations, after non-linear transformation
- Parameters: Matrices A_d, deep network T_d

$$\xi_d \sim \mathcal{N}(\xi_d; 0, \mathbb{I}) \ , d = 1, \ldots, D$$
$$z_D = G_D \xi_D$$
$$z_d = T_d(z_{d+1}) + A_d \xi_d \ , \ d = 1, \ldots, L$$
$$x \sim P(x; T_0(z_1))$$

- All parameters θ, prior $p(\theta) = \mathcal{N}(\theta; 0, \kappa \mathbb{I})$

Deep Latent Gaussian Models

- Two equivalent forms, intractable marginalization and inference

$$p(\mathsf{x}, \mathsf{z}, \theta) = p(\mathsf{x}|\mathsf{z}_1, \theta) \left[\prod_{d=1}^{D-1} p_d(\mathsf{z}_d|\mathsf{z}_{d+1}, \theta) \right] p(\mathsf{z}_D|\theta) p(\theta)$$

$$p(\mathsf{x}, \xi, \theta) = p(\mathsf{x}|\mathsf{z}_1(\xi_1, \ldots, \xi_D), \theta) p(\theta) \prod_{d=1}^{D} \mathcal{N}(\xi_d; 0, \mathbb{I})$$

- Main Ideas
  - Use an inference model, contrast with wake-sleep algorithm
  - Stochastic backpropagation, closed form for Gaussians
  - General cases
    - Use REINFORCE like MC estimate
    - Use re-parametrization, 'coordinate transformation'

References

- D. Kingma, M. Welling, Auto-Encoding Variational Bayes, ICLR, 2014.
- D. Rezende, S. Mohamed, D. Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models, ICML, 2014.
- D. Kingma and M. Welling, An Introduction to Variational Autoencoders, FTML, 2019.
- J. Paisley, D. Blei, M. Jordan, Variational Bayesian Inference with Stochastic Search, ICML, 2012.