Expressive VAEs

Instructor: John Thickstun

Discussion Board: Available on Ed

Zoom Link: Available on Canvas

Instructor Contact: thickstn@cs.washington.edu

Course Webpage: https://courses.cs.washington.edu/courses/cse599i/20au/

Variational Autoencoders

- Generative model $p_\theta(x, z) = p_\theta(x|z)r(z)$. Learn $p_\theta(x) \approx p(x)$, where

$$p_\theta(x) = \mathbb{E}_{z\sim r}[p_\theta(x|z)] = \int_{\mathcal{Z}} p_\theta(x|z)r(z)\,dz.$$

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) = \arg\max_\theta \sup_{q_\varphi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\varphi(\cdot|x)}} \log \frac{p_\theta(x, z)}{q_\varphi(z|x)}.$$

- Modeling choices: prior $r(z)$, likelihood $p_\theta(x|z)$, and proposal $q_\varphi(z|x)$.

Evaluating a VAE

- Report the ELBO? That's pessimistic!

- Use an importance sampling estimator:

$$\log p_\theta(x) = \log \mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \left[\frac{p_\theta(x,z)}{q(z|x)} \right] \approx \log \frac{1}{M} \sum_{i=1}^{M} \frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}, \text{ where } z_i \sim q(z|x).$$

- Consistent estimator of the log-likelihood as $M \to \infty$.

- The numerically stable version:

$$\log p_\theta(x) \approx \log \sum_{i=1}^{M} \exp\left(\log p_\theta(x|z_i) + \log p(z_i) - \log q_\phi(z_i|x)\right) - \log M.$$

Differential Entropy

- What is the negative log-likelihood of continuous data?

$$-\mathop{\mathbb{E}}_{x\sim p}\log p_\theta(x) \approx -\sum_{i=1}^{n}\log p_\theta(x_i), \text{ where } x_i \sim p.$$

- It is a monte carlo estimate of the continuous cross entropy

$$-\mathop{\mathbb{E}}_{x\sim p}\log p_\theta(x) = -\mathop{\mathbb{E}}_{x\sim p}\log p(x) - \mathop{\mathbb{E}}_{x\sim p}\log\frac{p_\theta(x)}{p(x)} = H(p) + D(p \parallel p_\theta).$$

- Differential entropy H(p) is a strange quantity. E.g. it can be negative!

- Quantitative empirical work tends to report results for discrete datasets.

- Use a discrete cross entropy loss instead of MSE reconstruction loss.

Variational Autoencoders

- Generative model $p_\theta(x, z) = p_\theta(x|z)r(z)$. Learn $p_\theta(x) \approx p(x)$, where

$$p_\theta(x) = \mathbb{E}_{z\sim r}[p_\theta(x|z)] = \int_{\mathcal{Z}} p_\theta(x|z)r(z)\,dz.$$

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) = \arg\max_\theta \sup_{q_\varphi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\varphi(\cdot|x)}} \log \frac{p_\theta(x, z)}{q_\varphi(z|x)}.$$

- Modeling choices: prior $r(z)$, likelihood $p_\theta(x|z)$, and proposal $q_\varphi(z|x)$.

Sharpening the ELBO

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_{\theta} \mathop{\mathbb{E}}_{x\sim p} \log \mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)} = \arg\max_{\theta} \sup_{q_\phi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\phi(\cdot|x)}} \log \frac{p_\theta(x,z)}{q_\phi(z|x)}.$$

- Use monte carlo estimates of the expectation (minibatch size M):

$$\mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \approx \sum_{i=1}^{M} \log \frac{p_\theta(x,z_i)}{q_\phi(z_i|x)}, \text{ where } z_i \sim q_\phi(\cdot|x).$$

- Importance-Weighted Autoencoders (IWAE):

$$\log \mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)} \approx \log \sum_{i=1}^{M} \frac{p_\theta(x|z_i)}{q_\phi(z_j|x)}, \text{ where } z_i \sim q_\phi(\cdot|x).$$

Sharpening the ELBO

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_{\theta} \mathop{\mathbb{E}}_{x\sim p} \log \mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)} = \arg\max_{\theta} \sup_{q_\phi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\phi(\cdot|x)}} \log \frac{p_\theta(x,z)}{q_\phi(z|x)}.$$

- Use monte carlo estimates of the expectation (minibatch size M):

$$\mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \approx \sum_{i=1}^{M} \log \frac{p_\theta(x,z_i)}{q_\phi(z_i|x)}, \text{ where } z_i \sim q_\phi(\cdot|x).$$

- Importance-Weighted Autoencoders (IWAE):

$$\log \mathop{\mathbb{E}}_{z\sim q_\phi(\cdot|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)} \approx \log \sum_{i=1}^{M} \frac{p_\theta(x|z_i)}{q_\phi(z_j|x)}, \text{ where } z_i \sim q_\phi(\cdot|x).$$

Variational Autoencoders

- Generative model $p_\theta(x, z) = p_\theta(x|z)r(z)$. Learn $p_\theta(x) \approx p(x)$, where

$$p_\theta(x) = \mathbb{E}_{z\sim r}[p_\theta(x|z)] = \int_{\mathcal{Z}} p_\theta(x|z)r(z)\, dz.$$

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) = \arg\max_\theta \sup_{q_\varphi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\varphi(\cdot|x)}} \log \frac{p_\theta(x, z)}{q_\varphi(z|x)}.$$

- Modeling choices: prior $r(z)$, likelihood $p_\theta(x|z)$, and proposal $q_\varphi(z|x)$.

- Any two can be Gaussian without hurting expressivity, but not all three!

Expressive Conditional Likelihood

- The Gaussian VAE has a conditionally independent likelihood:

$$p_\theta(x|z) = \mathcal{N}(x; g_\theta(z), \sigma^2 I) = \prod_{i=1}^{|\mathcal{X}|} \mathcal{N}(x_i|g_{\theta,i}(z), \sigma^2).$$

- Dependencies between coordinates/pixels x_i must be captured by z.

- Replace with an expressive autoregressive likelihood using NADE?

$$p_\theta(x|z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}, z).$$

PixelVAE

- Replace with an expressive autoregressive likelihood using NADE?

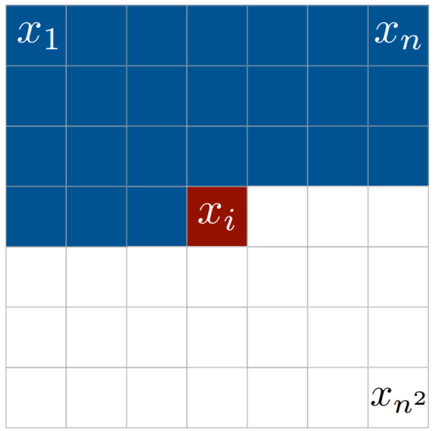
$$p_\theta(x|z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}, z).$$

- Sampling is expensive! $O(|\mathcal{X}|)$ serial calls to the model.

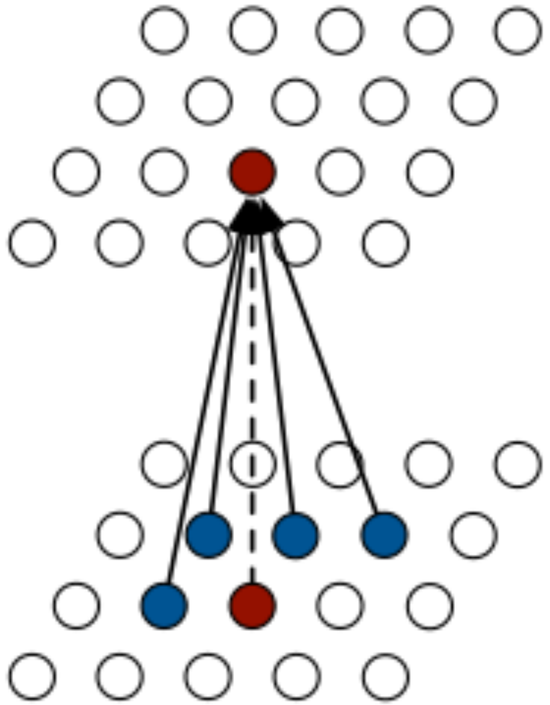
- $O(1)$ serial calls for a conditionally independent decoder.

An AR Models for Images

- Assign an (arbitrary) order the pixels, e.g. left-to-right, top-to-bottom:



- Does order matter?
- Doesn't seem so.
- Mask the observations so $g_\theta : (\mathcal{X}, \mathcal{Z}) \to \mathcal{X}$ only sees the history.
- I.e. $g_{\theta,i}(x_1, \ldots, x_{n \times n}) = g_{\theta,i}(x_1, \ldots, x_{<i})$.



Masked Convolutions

van den Oord et. al., ICML 2016

PixelCNN

- Why even bother with latent codes and the ELBO?

$$p_\theta(x|z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}, z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}).$$

- Directly learn an autoregressive model over the observed data.

- This is valid and still achieves SOTA log-likelihoods for tiny images.

- Poor scaling to large images (sampling time; long-range dependencies).

Posterior Collapse

- Why even bother with latent codes and the ELBO?

$$p_\theta(x|z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}, z) = \prod_{i=1}^{|\mathcal{X}|} p_\theta(x_i|x_{<i}).$$

- The ELBO is an lower bound, whereas AR likelihoods are exact:
  - Modeling the distribution with latent codes comes at a cost.
  - No comparable cost for modeling with sequence history.

- Optimization will result in $q_\phi(z|x) = r(z), p_\theta(x|z) = p_\theta(x)$ (posterior collapse).

5-Minute Break

Variational Autoencoders

- Generative model $p_\theta(x, z) = p_\theta(x|z)r(z)$. Learn $p_\theta(x) \approx p(x)$, where

$$p_\theta(x) = \mathbb{E}_{z\sim r}[p_\theta(x|z)] = \int_{\mathcal{Z}} p_\theta(x|z)r(z)\, dz.$$

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) = \arg\max_\theta \sup_{q_\varphi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\varphi(\cdot|x)}} \log \frac{p_\theta(x, z)}{q_\varphi(z|x)}.$$

- Modeling choices: prior $r(z)$, likelihood $p_\theta(x|z)$, and proposal $q_\varphi(z|x)$.

- Any two can be Gaussian without hurting expressivity.

Expressive Prior

- Generative model $p_{\theta,\psi}(x,z) = p_\theta(x|z)r_\psi(z)$. Learn $p_{\theta,\psi}(x) \approx p(x)$, where

$$p_\theta(x) = \mathbb{E}_{z\sim r_\psi}[p_\theta(x|z)] = \int_{\mathcal{Z}} p_\theta(x|z)r_\psi(z)\,dz.$$

- Estimate the MLE using the ELBO:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) = \arg\max_{\theta,\psi} \sup_{q_\phi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\phi(\cdot|x)}} \log \frac{p_\theta(x|z)r_\psi(z)}{q_\phi(z|x)}.$$

- No tricks needed to optimize the prior.

- How to model the prior? An autoregressive model? Slow to sample.

Inference Suboptimality

- No reason for the true posterior distribution $p_\theta(z|x)$ to be Gaussian!

- The ELBO is loose no matter how well we optimize the proposal:

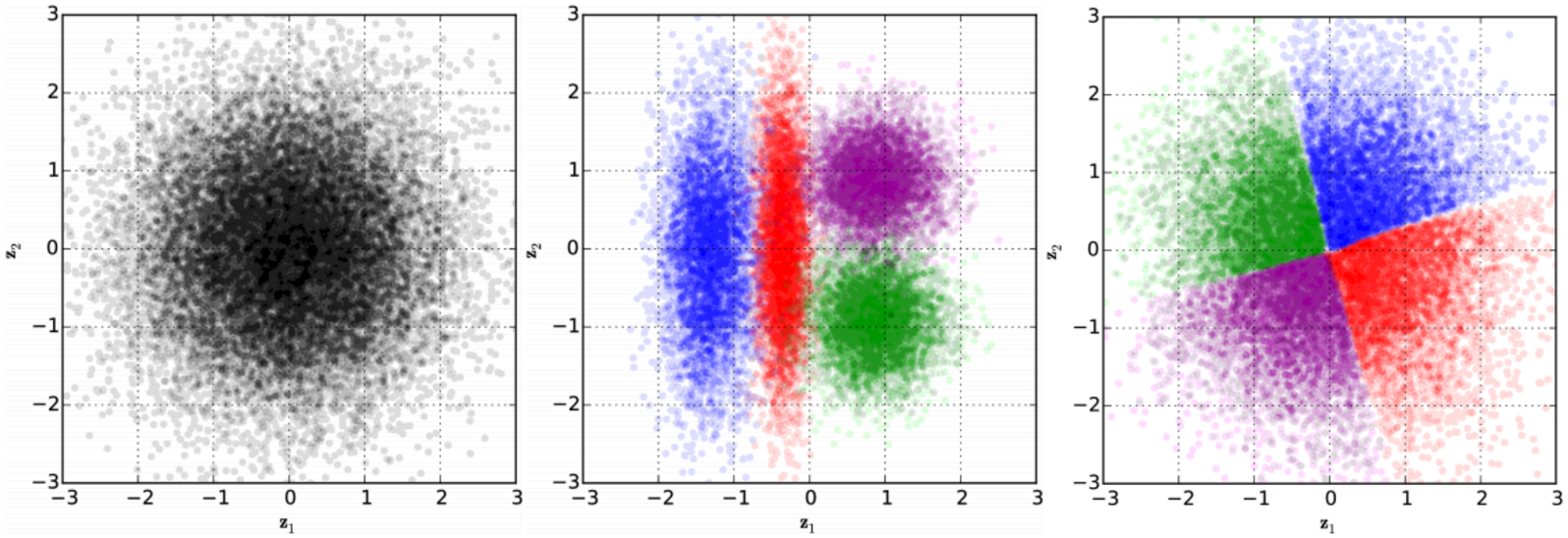
$$\mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) > \sup_{q_\phi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\phi(\cdot|x)}} \log \frac{p_\theta(x,z)}{q_\phi(z|x)}.$$

- With a Gaussian proposal family, we won't be able to learn the MLE:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p} \log p_\theta(x) \neq \arg\max_\theta \sup_{q_\phi} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q_\phi(\cdot|x)}} \log \frac{p_\theta(x,z)}{q_\phi(z|x)}.$$

Visualizing Suboptimality

- Two VAE's fit to a dataset with four datapoints.
- Each color cloud corresponds to the proposal distribution for a datapoint.



(a) Prior distribution (b) Posteriors in standard VAE (c) Posteriors in VAE with IAF

Kingma et. al., Neurips 2016

Expressive Proposal

- Use an autoregressive proposal?

$$q_\phi(z|x) = \prod_{i=1}^{|\mathcal{Z}|} q_\phi(z_i|z_{<i}, x).$$

- The usual problem: sampling is slow.

- We need to sample *during training.*

Normalizing Flows

- Build a series of transformations of our initial proposal $\mathbf{z}_0 \sim q_\phi(\cdot|x)$.
- Let $g_s : \mathcal{Z} \to \mathcal{Z}$ and define $\mathbf{z}_t = g_t \circ \cdots \circ g_1(\mathbf{z}_0)$.
- The log-density of the pushfoward distribution on $\mathbf{z}_t$ is given by

$$\log q_t(\mathbf{z}_t) = \log q_0(\mathbf{z}_0) - \sum_{s=1}^{t} \operatorname{logdet} \left(\frac{\partial g_s(\mathbf{z}_{t-1})}{\partial \mathbf{z}_{t-1}} \right).$$

- Choose functions g_s so that $\operatorname{logdet} \left(\frac{\partial g_s(\mathbf{z}_{t-1})}{\partial \mathbf{z}_{t-1}} \right)$ is easy to calculate.