STA 4273: Minimizing Expectations

Lecture 6 - Variational Objectives II

Chris J. Maddison

University of Toronto

Announcements

- Working to get marks / feedback on the proposals by EOW.
- I'm interested in sharing your great work! Email me:
  - Can I share your slides on Quercus?
  - Can I share your slides on the course website?
  - Can I share your code notebook on Quercus?
  - Can I share your code notebook on the course website?
- Questions, comments, concerns?

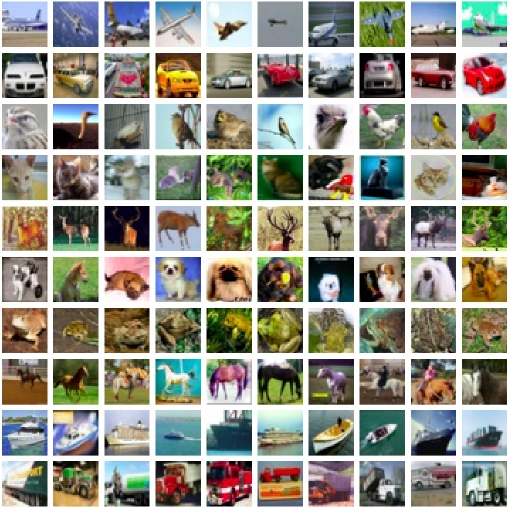
Modelling high-dimensional, multi-modal data

MNIST handwritten digit dataset.

(Vahdat and Kautz, 2020)

Modelling high-dimensional, multi-modal data

CIFAR-10 small natural image dataset.



(Krizhevsky, 2009)

Modelling high-dimensional, multi-modal data

CelebA large images of celebreties

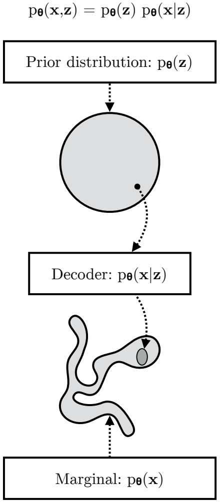
(Liu et al., 2015)

Variational autoencoders

1. Variational autoencoders (VAEs) are latent variables models for high dimensional data $\mathbf{x} \in \mathbb{R}^n$.
2. A latent variable model is specified in terms of a joint distribution between $\mathbf{x}$ and a latent variable $\mathbf{z} \in \mathbb{R}^m$ that factorizes as follows:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z})$$

3. Latent variable models are an expressive class, because the marginal $p_\theta(\mathbf{x})$ can be very complex due to the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ warping the probability mass of a simple prior $p_\theta(\mathbf{z})$.



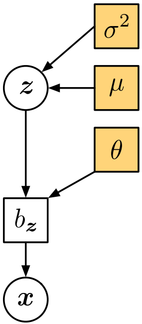
(Kingma and Welling, 2019)

Variational autoencoders—example

1. Consider the binary data case, $\mathbf{x} \in \{0,1\}^n$.
2. Consider a deep Gaussian latent variable model.

$$\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2 I)$$
$$\mathbf{x}_i \sim \text{Bernoulli}(b_{\mathbf{z},i}) \text{ indept.}$$

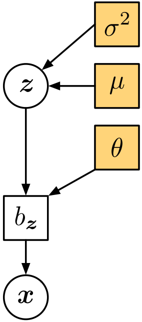
   where $b_{\mathbf{z}} = \mathcal{NN}_\theta(\mathbf{z})$ is computed using a neural network $\mathcal{NN}_\theta : \mathbb{R}^m \to [0,1]^n$ with parameters θ.
3. The marginal $p_\Theta(\mathbf{x})$ can be multimodal and expressive.



1. Let $\Theta = (\theta, \mu, \sigma^2)$. How can we do maximum likelihood over Θ in this model?
2. What we want is

$$\arg\max_{\Theta} \log p_{\Theta}(\mathbf{x})$$

   but $p_{\Theta}(\mathbf{x}) = \int p_{\Theta}(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$ is too expensive to compute.
3. The basic idea behind the variational autoencoder is to optimize a tractable variational lower bound on $\log p_{\Theta}(\mathbf{x})$, in fact the ELBO (Lecture 1)!



Evidence lower bound

Recall the evidence lower bound (ELBO)

$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi}\left[\log \frac{p_\Theta(\mathbf{z}, \mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \log p_\Theta(\mathbf{x}) - KL(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\Theta(\mathbf{z}|\mathbf{x}))$$

Where

- q_ϕ is a density in a parametric family of probability densities.
- The objective is called the ELBO, because:

$$\text{ELBO}(\Theta, \phi, \mathbf{x}) \leq \log p_\Theta(\mathbf{x})$$

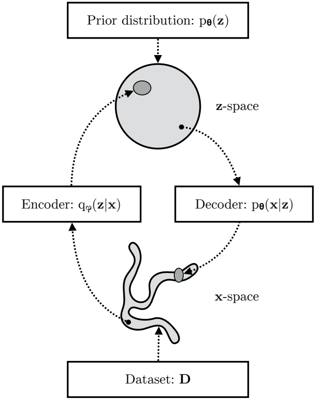
Idea: what if we optimized the ELBO in terms of Θ, ϕ?

Variational autoencoders

1. Approximate maximum likelihood for VAEs is carried out by introducing a approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$.
2. To fit a VAE, optimize ELBO using gradient ascent as a surrogate for the the marginal likelihood of $\mathbf{x}$,

$$\mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log \frac{p_\Theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$$

3. The key question is then how to estimate $\nabla_\Theta \text{ELBO}(\Theta,\phi,\mathbf{x})$ and $\nabla_\phi \text{ELBO}(\Theta,\phi,\mathbf{x})$



(Kingma and Welling, 2019)

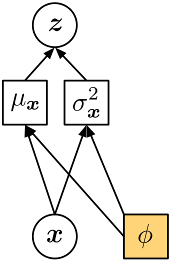
Variational autoencoders

1. In practice, q_ϕ is also implemented using neural network to make it more expressive.

$$\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{x}}, \mathrm{diag}(\sigma_{\mathbf{x}}^2))$$

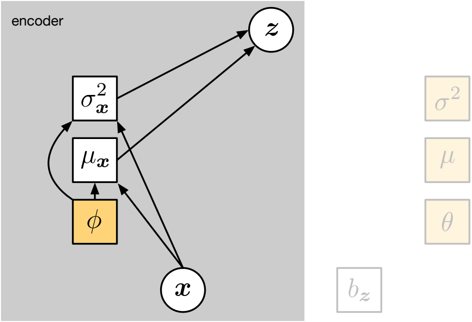
where $\mu_{\mathbf{x}}, \sigma_{\mathbf{x}}^2$ are computed using neural networks with parameters ϕ, as with $b_{\mathbf{z}}$.

2. OK, we defined both p_Θ and q_ϕ, but how can we estimate gradients of $\mathrm{ELBO}(\Theta, \phi, \mathbf{x})$?
3. Let's consider the SCG that simulates a realization of the ELBO.



SCG for the ELBO

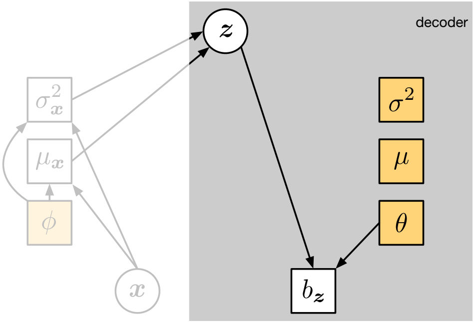
Let's first add the graph that samples from q_ϕ, called the encoder.



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

SCG for the ELBO

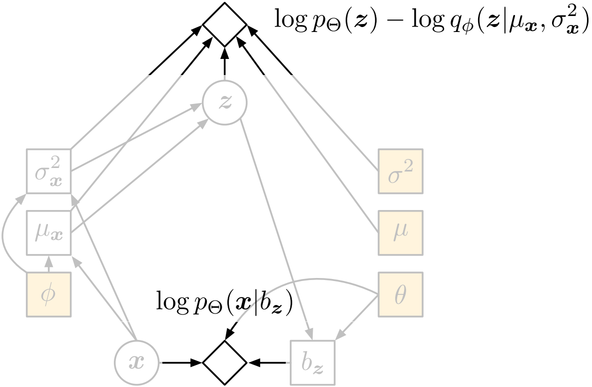
Now, a graph that computes the statistics of $p_\theta(\mathbf{x}|\mathbf{z})$, called the decoder.



$$\mathrm{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

SCG for the ELBO

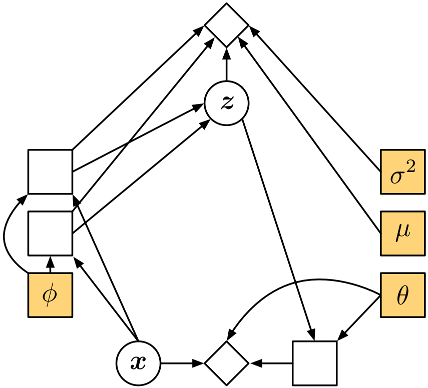
Finally, the graph that computes the losses.



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

SCG for the ELBO

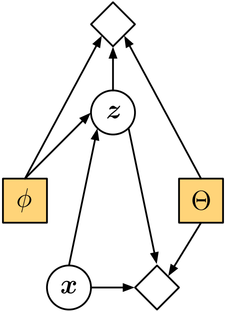
To optimize the expected losses over Θ, ϕ, consider the gradients that we want.



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi} \left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}) \right]$$

SCG for the ELBO

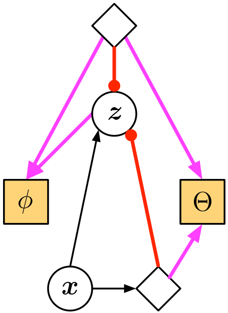
First, simplify. Goal: find all of the paths from loss nodes to orange nodes.



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi} [\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

SCG for the ELBO

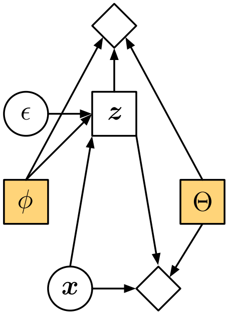
z blocks 2 paths. Can use score function est., but high variance.



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

SCG for the ELBO

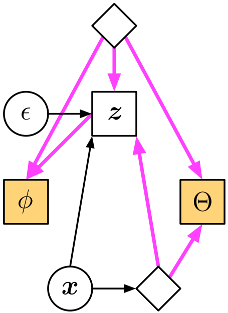
Luckily, we can reparameterize the graph with $\mathbf{z} = \sigma_{\mathbf{x}}\epsilon + \mu_{\mathbf{x}}$:



$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

SCG for the ELBO

Now we get pathwise gradients! Much lower variance!



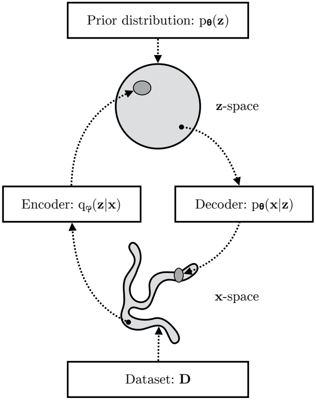
$$\text{ELBO}(\Theta, \phi, \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi} [\log p_\Theta(\mathbf{z}) + \log p_\Theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

Variational autoencoders—summary

1. A VAE is a latent variable model $p_\Theta(\mathbf{x}, \mathbf{z})$.
2. To fit a VAE,
   - introduce an approximate posterior $q_\phi(\mathbf{z}|\mathbf{z})$.
   - optimizing the ELBO using gradient ascent

$$\mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log \frac{p_\Theta(\mathbf{z}, \mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$$

   - compute ELBO gradients by reparameterizing a SCG that simulates the ELBO.



(Kingma and Welling, 2019)

Variational autoencoders

- Variational autoencoders can get quite elaborate.
- A (now old, but cool) example is the DRAW model (Gregor et al., 2015).
  - DRAW: A Recurrent Neural Network For Image Generation
- This is a time-series model that turns generation in an iterative process using attention.
- It is basically an elaborate VAE.

Variational bayes

- The idea of variational inference is applicable beyond latent variable models.
- We can use variational inference for the problem of Bayesian inference.
- Suppose we have a regression or classification task from inputs $\mathbf{x} \in \mathbb{R}^d$ to labels $\mathbf{y} \in \mathcal{Y}$. We can use a neural network with parameters $\mathbf{w} \in \mathbb{R}^n$ that parameterizes a distribution $p(\mathbf{y}|,\mathbf{w})$.
- Maximum likelihood corresponds to

$$\max_{\mathbf{w} \in \mathbb{R}^n} \log p(\mathbf{y}|\mathbf{x}, \mathbf{w})$$

Variational bayes

- Maximum likelihood is prone to overfitting, why not "be Bayesian"?
  - This course is not about statistical inference, so I don't want to get into pointless arguments about whether being Bayesian is correct.
  - Training multiple diverse models and averaging their predictions (ensembling) is a very effective technique for reducing variance (overfitting) in practice (and theory in some settings).
- Being Bayesian ultimately amounts to saying that you want to average over multiple parameter settings, instead of maximize. I.e., you want to use the following to predict:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{y}, \mathbf{x}) \, d\mathbf{w}$$

Variational bayes

- What the heck is $p(\mathbf{w}|\mathbf{y},\mathbf{x})$ and how do we get it?
- $p(\mathbf{w}|\mathbf{y},\mathbf{x}) \propto p(\mathbf{y}|\mathbf{x},\mathbf{w})p(\mathbf{w})$ is the "posterior" and it is determined by some choice of prior $p(\mathbf{w})$.
- The topic of Bayesian inference ultimately amounts to computing expectations w.r.t. $p(\mathbf{w}|\mathbf{y},\mathbf{x})$, and we can approximate it with variational inference! Variational bayes:

$$p(\mathbf{w}|\mathbf{y},\mathbf{x}) = \arg\max_q \mathbb{E}\left[\log \frac{p(\mathbf{y}|\mathbf{x},\mathbf{w})p(\mathbf{w})}{q(\mathbf{w}|\mathbf{y},\mathbf{x})}\right]$$

- Main idea is, we can use variational inference (and the techniques we've learned today) for more than just latent variable models.

Talks today

- Variational bayes for neural network parameters using the reparameterization trick (just like VAEs!).
- Variational bayes over the neural network *function space* using ideas from gradient estimation for implicit models.
- Optimizing variational objectives that are not the ELBO (KLs in the other direction).