Machine Learning and Optimization Lecture 26

- Review: KL-divergence of Gaussian distributions
- Latent generative models
- Variational Autoencoders (VAEs)
- Evidence Lower Bound (ELBO) and decomposition into negative reconstruction error and regularizer
- Optimizing the ELBO

Review

The KL divergence between two pdfs p and q on $\mathbb{R}^d$:

$$D_{KL}(p\|q) = \int_{\mathbb{R}^d} p(x) \log\left[\frac{p(x)}{q(x)}\right] dx$$

$$= \int_{\mathbb{R}^d} p(x) \log p(x)\, dx - \int_{\mathbb{R}^d} p(x) \ln q(x)\, dx$$

Facts:

- $D_{KL}(p\|q) \geq 0$ for any p, q
- $D_{KL}(p\|q) = 0$ iff $p = q$
- $D_{KL}(p\|q) \neq D_{KL}(q\|p)$ in general
- $D_{KL}(p\|q) = \infty$ if there exists a set with positive probability under p that has zero probability under q.

This implies that choosing q to minimize $D_{KL}(p\|q)$ will result in a q satisfying $\text{supp}(p) \subseteq \text{supp}(q)$

In particular, if p and q are Gaussians,

$$p(x) = \frac{1}{(2\pi)^{d/2} \det(\Sigma_p)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_p)\Sigma_p^{-1}(x-\mu_p)\right)$$

$$q(x) = \frac{1}{(2\pi)^{d/2} \det(\Sigma_q)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_q)\Sigma_q^{-1}(x-\mu_q)\right)$$

then

$$D_{KL}(p \| q) =$$

$$\frac{1}{2}\left[\operatorname{tr}(\Sigma_q^{-1}\Sigma_p) - d + (\mu_q-\mu_p)^T\Sigma_q^{-1}(\mu_q-\mu_p) + \ln\left(\frac{\det\Sigma_q}{\det\Sigma_p}\right)\right]$$

has a closed form.

In particular, if $\Sigma_p = \mathrm{Diag}(\sigma_1^2, \dots, \sigma_d^2)$ and $\mu_q = 0$, $\Sigma_q = I_d$, then

$$D_{KL}(p \| q) = \frac{1}{2}\left[\mathrm{Tr}(\Sigma_p) - d + (-\mu_p)^T I (-\mu_p) + \ln\left(\frac{1}{\det(\Sigma_p)}\right)\right]$$

$$= \frac{1}{2}\left[\left(\sum_{i=1}^{d} \sigma_i^2\right) - d + \|\mu_p\|_2^2 - \ln\left(\prod_{i=1}^{d} \sigma_i^2\right)\right]$$

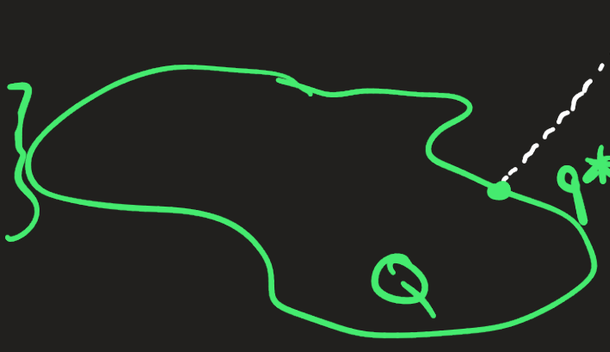
$$= \frac{1}{2}\sum_{i=1}^{d}\left(\sigma_i^2 - 1 + (\mu_p)_i^2 - \ln(\sigma_i^2)\right)$$

We will find this expression useful.

Variational Approximation

Given a reference distribution p, we are often interested in approximating it with the "closest" distribution in a class $\mathcal{Q}$.

$$\mathcal{Q} = \{ q_\phi(x) \mid \phi \in \mathbb{R} \}$$



A natural choice is

$$q^* = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \; D_{KL}(p \,\|\, q) \quad \alpha\text{-}\mathbb{E}_{x \sim p} \ln q(x)$$

this is MLE!

Another choice that can be algorithmically more attractive is

$$q^* = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \; D_{KL}(q \,\|\, p) \quad \alpha\text{-}\mathbb{E}_{x \sim q} \ln p(x)$$

commonly called variational approximation

VAEs in a picture

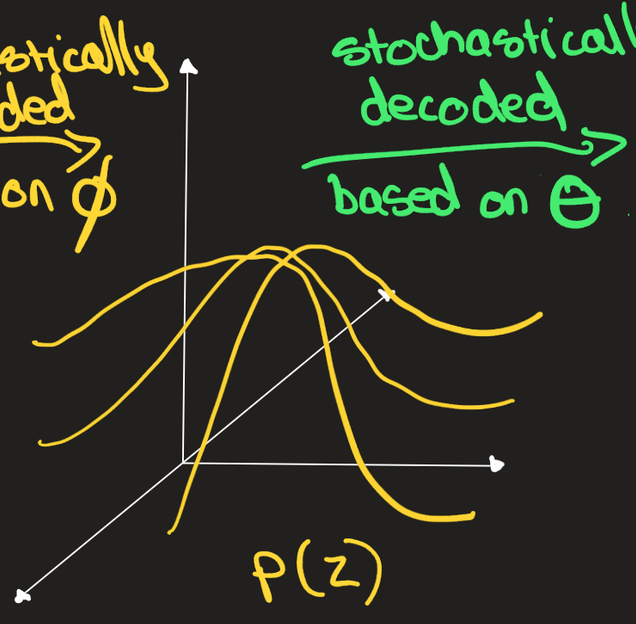
"Like autoencoders, but stochastic"

Goal: learn ϕ and θ to balance between accurately reconstructing the learning data and learning a "meaningful" latent space

x

Data Space
$\mathbb{R}^s$

stochastically encoded
based on ϕ



Latent Space
$\mathbb{R}^d$

stochastically decoded
based on θ

Data Space
$\mathbb{R}^s$

Latent variable generative model

Consider a dataset $X = \{x_1, \dots, x_n\}$, where each instance is s-dimensional and are i.i.d samples from p_{data}.

Assume there exist n corresponding latent variables (unseen!) $Z = \{z_1, \dots, z_n\}$ where each variable is d-dimensional ($d \ll s$), and X and Z satisfy the generative model

$z_i \sim p(z)$ and $x_i | z_i \sim p_\theta(x|z_i)$, so $p_\theta(x,z) = p_\theta(x|z) \cdot p(z)$.

In the case of variational autoencoders (VAEs), our goal is to learn a probabilistic decoder $p_\theta(x|z)$ so that

$$p_{data}(x) = p_\theta(x) = \int p_\theta(x,z)\, dz = \int p_\theta(x|z)\, p(z)\, dz$$

↑ Intractable to compute!

Notice that by Bayes' rule, if we learn the decoder then the matching encoder is given by

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

Ex

Let $x \in [0,1]^{28\times 28}$ be an image and $z \in \mathbb{R}^{10}$ be a latent code, then we can take the decoder to be parameterized by

$$p_\theta(x|z) = \mathcal{N}(\text{CNN}_\theta(z), I)$$

where $\text{CNN}_\theta(z)$ denotes a transposed convolution architecture

A common choice for $p(z)$ is $\mathcal{N}(0, I_d)$

Tasks we want to do:

1) Learn a good decoder, in the sense that

$$p_\theta(x) = \int p_\theta(x|z)p(z)dz = p_{data}(x)$$

MLE is intractable because the integral $p_\theta(x)$ only be approximated numerically

2) Learn a good encoder $q_\phi(z|x)$, in the sense that

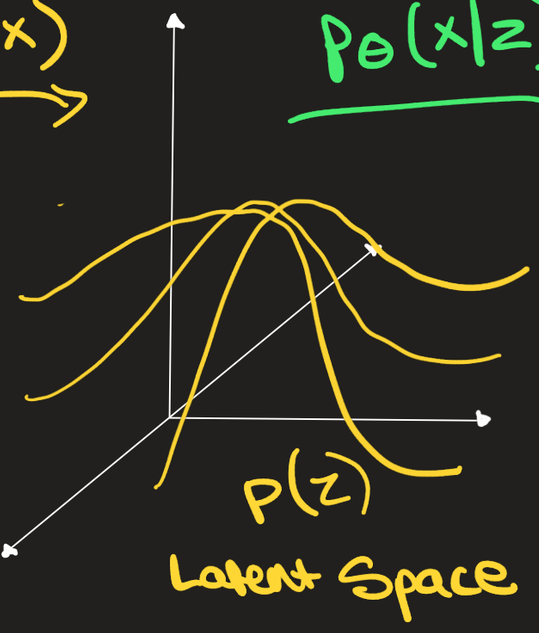
$$q_\phi(z|x) \approx p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)} = \frac{p_\theta(x|z)p(z)}{p_\theta(x)}$$

We learn q_ϕ because again, computing $p_\theta(x)$ is intractable. This approximation choice is what defines VAEs

VAE Architecture

x

$q_\phi(z|x)$



Latent Space

$p_\theta(x|z)$

Data Space
$\mathbb{R}^s$

Data Space
$\mathbb{R}^s$

$$q_\phi(z|x) = \mathcal{N}\left(\mu_\phi(x), \text{Diag}(\sigma_\phi(x))\right)$$

Neural networks parameterized by ϕ

$$p_\theta(x|z) = \mathcal{N}\left(\text{CNN}_\theta(z), \sigma^2 \cdot I\right)$$

or

$$p_\theta(x|z) = p_\theta(x_0|z) \prod_{i=1}^{s} p_\theta(x_i|z, x_0, \ldots x_{i-1})$$

Neural networks parameterized by θ

Example use-cases:

- Given an image x, we sample z from $q_\phi(z|x)$ and then sample x' from $p_\theta(x|z)$ to get a similar, but random, image

- To generate a random image from p_{data}, we sample z from the prior $p(z)$, then sample an image from $p_\theta(x|z)$

Now we consider how to learn the encoder and decoder. To begin with, assume we have learn a decoder $p_\theta(x|z)$. We approximate the intractable encoder $p_\theta(z|x)$ variationally, by finding the closest distribution in some parameterized class,

$$\hat{\phi} = \arg\min_{\phi} \mathbb{E}_{x \sim p_{data}} D_{KL}(q_\phi(z|x) \,\|\, p_\theta(x|z))$$

It turns out that this can be solved without need to compute (the intractable) $p_\theta(x)$!

Ex for the image example before, we can take

$q_\phi(z|x) = \mathcal{N}(\mu(x;\phi), \Sigma(x;\phi))$ where μ and Σ are scalar-valued CNNs and learn the optimal ϕ

Notice

$$(*)\ D_{KL}(q_\phi(z|x) \| p_\theta(z|x)) = \int q_\phi(z|x) \ln \frac{q_\phi(z|x)}{p_\theta(z|x)}\, dz$$

where $p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)} = \frac{p_\theta(x|z)\,p(z)}{p_\theta(x)}$, so

$$(*) = \int q_\phi(z|x) \ln \frac{q_\phi(z|x)}{p(z)}\, dz + \int q_\phi(z|x) \ln \frac{p_\theta(x)}{p_\theta(x|z)}\, dz$$

$$= D_{KL}(q_\phi(z|x) \| p(z)) - \int q_\phi(z|x) \ln p_\theta(x|z)\, dz + \ln p_\theta(x)$$

$$= D_{KL}(q_\phi(z|x) \| p(z)) - \mathbb{E}_{z\sim q_\phi(z|x)} \ln p_\theta(x|z) + \ln p_\theta(x)$$

We have established that

$$D_{KL}(q_\phi(z|x) \| p_\theta(z|x)) = \ln p_\theta(x) - \Big[\underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \ln p_\theta(x|z) - D_{KL}(q_\phi(z|x) \| p(z))}_{\text{Evidence Lower Bound} := \mathcal{L}_{ELBO}(\phi, \theta; x)} \Big]$$

Thus, we learn a good encoder by solving

$$\hat{\phi} = \arg\max_{\phi} \mathbb{E}_{x \sim p_{data}} \mathcal{L}_{ELBO}(\phi, \theta; x)$$

Interpreting the ELBO

$$\mathcal{L}_{ELBO}(\phi, \theta; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \ln p_\theta(x|z)}_{\text{reconstruction term } \mathcal{L}_{AE}} - \underbrace{D_{KL}(q_\phi(z|x) \,\|\, p(z))}_{\text{regularizer } \mathcal{L}_{reg}}$$

Reconstruction term

The input x gets encoded to the distribution $q_\phi(z|x)$. The decoder then reconstructs an image by sampling a latent vector from $q_\phi(z|x)$. Ideally this process should result in an image like x. We quantify this by

$$\mathbb{E}_{z \sim q_\phi(z|x)} \ln p_\theta(x|z) = \int q_\phi(z|x) \ln p_\theta(x|z) \, dz$$

Regularizer

For the latent vectors to be meaningful across multiple inputs, $q_\phi(z|x)$ cannot be too idiosyncratic to a particular x. This term regularizes the encodings $q_\phi(z|x)$ to stay close to prior $p(z)$

arbitrarily powerful $p_\phi(z|x)$

w/o regularization

w/ regularization

House

Cat

Dog

Horse

?

?

?

$p(z|House)$

$p(z|house)$

$p(z|house)$

not cute

animate

size

$p(z|cat)$

$p(z|dog)$

$p(z|dog)$

$p(z|horse)$

$p(z|cat)$

Getting a good decoder

The ideal decoder maximizes the MLE

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

but this is intractable. However, we have shown that

$$D_{KL}(q_\phi(z|x) \,\|\, p(z)) = \ln p_\theta(x) - \mathcal{L}_{ELBO}(\phi, \theta; x),$$

which implies

$$\ln p_\theta(x) \geq \mathcal{L}_{ELBO}(\phi, \theta; x)$$

To learn a good decoder, assume we have an encoder $q_\phi(z|x)$, then since the MLE is intractable, we maximize $\mathcal{L}_{ELBO}(\phi, \theta)$, the lower bound on $\ln p_\theta(x)$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} - \mathbb{E}_{x \sim p_{data}(x)} \mathcal{L}_{ELBO}(\phi, \theta; x)$$

Thus, to learn a VAE, we: 0) fix a prior $p(z)$

1) fix the architecture for the decoder $p_\theta(x|z)$
2) fix the architecture for the encoder $q_\phi(z|x)$
3) Learn the encoder and decoder by solving

$$\hat{\phi}, \hat{\theta} = \underset{\phi}{\operatorname{argmin}} \underset{\theta}{\operatorname{argmin}} - \mathbb{E}_{x \sim p_{data}} \mathcal{L}_{ELBO}(\phi, \theta; x)$$

Ex

To learn a VAE for MNIST images, take

$$p(z) = \mathcal{N}(0, I_d)$$

$$p_\theta(x|z) = \mathcal{N}\left(\text{tCNN}_\theta(z), I_s\right)$$

and

$$q_\phi(z|x) = \mathcal{N}\left(\underbrace{\text{CNN}_\phi(x)}_{\mu}, \text{Diag}\left(\underbrace{\text{CNN}_\phi(x)}_{\sigma^2}\right)\right)$$

Then

$$\mathcal{L}_{AE} = \mathbb{E}_{z\sim q_\phi(z|x)} \underbrace{\frac{1}{2}\|\text{tCNN}_\theta(z) - x\|_2^2}$$

$$\mathcal{L}_{Reg} = D_{KL}(q_\phi(z|x) \,\|\, p(z))$$

$$= \frac{1}{2}\sum_{i=1}^{d}\left[\sigma_i^2(x) - 1 + \mu_i(x)^2 + \ln \sigma_i^2(x)\right]$$

Important question: what do we lose by maximizing the ELBO instead of the evidence?

- We have shown $D_{KL}(q_\phi(z|x) \| p_\theta(z|x)) = \ln p_\theta(x) - \mathcal{L}_{ELBO}(\phi;\theta;x)$
so the closer q_ϕ is to the true posterior $p_\theta(z|x)$,
the closer the ELBO is to the evidence

- If the posterior $p_\theta(z|x)$ is always in the variational family q_ϕ for some ϕ, then we have

$$\ln p_\theta(x) = \mathcal{L}_{ELBO}(\phi, \theta; x)$$

So if $p_{data} = p_\theta$ for some θ, this procedure can learn θ so $p_{data}(x) = p_\theta(x)$.

Suggests using high capacity NNs for the encoder and decoder

How do we solve

$$\hat{\phi}, \hat{\theta} = \underset{\phi,\theta}{\operatorname{argmax}} \mathbb{E}_{x \sim p_{data}} \mathcal{L}_{ELBO}(\phi, \theta; x)$$

in practice?

We use gradient ascent-ascent. To facilitate taking the gradient w.r.t. ϕ, we take

$$q_\phi(z|x) = \mathcal{N}(\mu(x;\phi), \operatorname{Diag}(\sigma^2(x;\phi)))$$

where $\sigma^2(x;\phi)$ is a vector with nonnegative entries

and we take $p(z) = \mathcal{N}(0, I_d)$

With these choices,

$$\mathcal{L}(\phi,\theta;x) = \mathbb{E}_{z\sim q_\phi(z|x)} \ln p_\theta(x|z) - D_{KL}(q_\phi(z|x)\,\|\,p(z))$$

$$= \mathbb{E}_{z\sim q_\phi(z|x)} \ln p_\theta(x|z) - \frac{1}{2}\sum_{i=1}^{d}\left[\sigma_i^2(x;\phi) - 1 + \mu_i(x;\phi)^2 + \ln\sigma_i^2(x;\phi)\right]$$

Naively differentiating through the reconstruction term requires MCMC approaches to estimate the integral.
Instead, we use the "reparameterization trick"

$$\mathbb{E}_{z\sim q_\phi(z|x)} f(z) = \int q_\phi(z|x) f(z)\,dz$$

$$= \int p(z) f(\sigma(x;\phi)z + \mu(x;\phi))\,dz$$

$$= \mathbb{E}_{z\sim \mathcal{N}(0,I)} f(\sigma(x;\phi)z + \mu(x;\phi))$$

Notice we scale each coordinate by the standard deviation

Thus we compute a stochastic gradient g_ϕ s.t.

$$\mathbb{E} g_\phi = \nabla_\phi \mathbb{E}_{x \sim p_{data}} \mathcal{L}(\phi, \theta; x)$$

by sampling x from the training data and z from $p(z)$ and taking

$$g_\phi = \nabla_\phi \ln p_\theta(x | \sigma(x;\phi) z + \mu(x;\phi))$$

$$= -\frac{1}{2} \sum_{i=1}^{d} \nabla_\phi \left[\sigma_i^2(x;\phi) + \mu_i(x;\phi)^2 + \ln \sigma_i^2(x;\phi) \right]$$

Similarly,

$$g_\theta = \nabla_\theta \ln p_\theta(x | \sigma(x;\phi) z + \mu(x;\phi))$$

satisfies

$$\mathbb{E} g_\theta = \nabla_\theta \mathbb{E}_{x \sim p_{data}} \mathcal{L}(\phi, \theta; x)$$