CSDS 440: Machine Learning

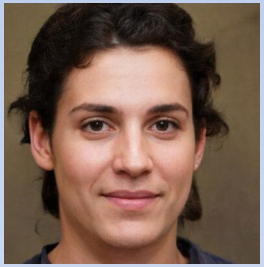
Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

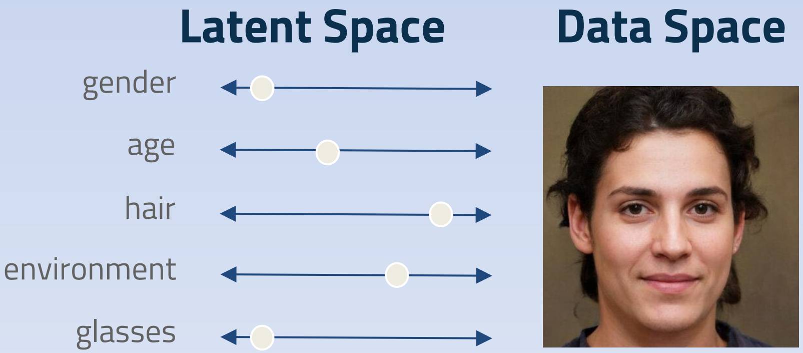
High dimensional Generative Models

- Suppose we wish to generate an image of a face
- This is hard!
- These are samples from a VERY high dimensional distribution
- And, the “axes” are not independent



Latent Variable Models

- To make the problem easier, we introduce "latent variables" z
  - These are hidden features which capture (hopefully independent) abstractions that constrain the space of images



Latent Space

Data Space

gender

age

hair

environment

glasses

Maximizing Likelihood

- Observe

$$p(X) = \int p(X \mid z)\, p(z)\, dz$$

- To train a model, want to maximize LHS as before

1. What should z be?
2. How to compute the $p(X)$ above?

First clever idea

- We have no idea what z could be
- Let us just sample z from $N(\mathbf{0}, \boldsymbol{I})$ and use a nonlinear function to *transform* this input into the output we need
  - Does such a function exist?
  - In many cases yes! under "compatibility" conditions for $p(X)$ and sufficiently rich nonlinear transformations

And so

- We'll choose a trainable family $f_\theta(z)$, typically a neural network

- With this choice, $p(X \mid z) = N(f_\theta(z), \sigma^2 I)$

Evaluating Likelihood

$$p(X) = \int p(X \mid z) p(z) dz, z \sim N(0, I)$$

$$\approx \frac{1}{n} \sum_i p(X \mid z_i)$$

- Unfortunately, in high dimensions, most $p(X \mid z_i)$ will be near zero, so this is going to be VERY inefficient

Second key idea

- What if we had a function $Q(z \mid X)$, that could return a distribution over those z's that are likely to produce X?

- Then maybe we could use $E_{z\sim Q}\, p(X \mid z)$ to get a good approximation to the likelihood?

Aside: Kullback-Liebler divergence

- One way to measure the “dissimilarity” between two distributions

$$D(X(z)\,||\,Y(z)) = E_{z\sim X}\left(\log\left(X(z)\right) - \log\left(Y(z)\right)\right)$$

Relationship between $E_{z\sim Q}\, p(X \mid z)$ and $p(X)$

$$D(Q(z \mid X) \,\|\, p(z \mid X))$$

$$= E_{z\sim Q}\left(\log\left(Q(z \mid X)\right) - \log\left(p(z \mid X)\right)\right)$$

$$= E_{z\sim Q}\left(\log\left(Q(z \mid X)\right) - \log\left(p(X \mid z)\right) - \log\left(p(z)\right)\right)$$

$$+\log\left(p(X)\right)$$

So

$$\log\left(p(X)\right) - D(Q(z \mid X) \,\|\, p(z \mid X)) =$$

$$E_{z\sim Q}\left(\log\left(p(X \mid z)\right)\right) - D\left(Q(z \mid X) \,\|\, p(z)\right)$$

Observations

- If we can find a good Q, the LHS $\approx p(X)$
- The RHS can be optimized via SGD! (w/suitable choices)
  - Not the LHS due to $p(z|X)$
- The RHS is called an "encoder-decoder" architecture
  - Q is given X and is "encoding" it into z
  - p (through the unknown f introduced before) will take z and "decode" it into X

Optimizing the RHS

- What to choose for $Q(z \mid X)$?
- Since the prior and likelihood are Gaussian, set $Q(z \mid X)$=$N(\mu_{\varphi}(X)$,$\Sigma_{\varphi}(X))$
  - In this case, this will be a single ANN φ that takes X as input and outputs μ and Σ
- With this choice, the second term on the RHS can be computed in closed form

Second term

$$D\left(Q(z \mid X) \,||\, p(z)\right) =$$

$$\frac{1}{2}\left[tr(\Sigma(X)) + \mu(X)^T \mu(X) - k - \log\left(\det\left(\Sigma(X)\right)\right)\right]$$

Trace

Dimensionality of z

Determinant

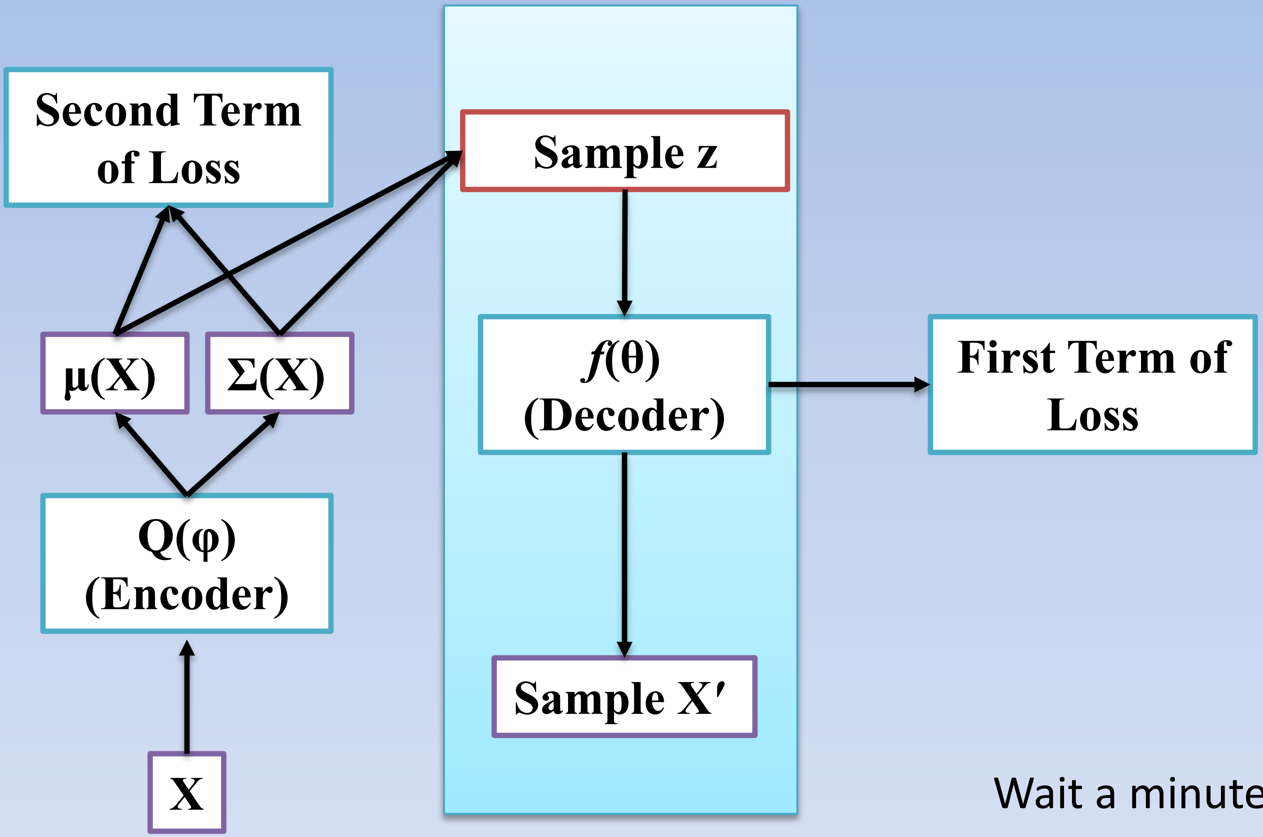
Note: μ, Σ are outputs of Q

First term

$$E_{z\sim Q}\left(\log\left(p(X\mid z)\right)\right)?$$

- Do we need to sample many times? That would be a problem…
- Third clever idea: *One* z sample can be enough!!
  - Why? When we do SGD, every time we run through an example x_i, we will resample z_i, so in the limit of enough epochs the stochastic gradient should converge to the true gradient under expectation!

The Variational Auto-Encoder (VAE) architecture



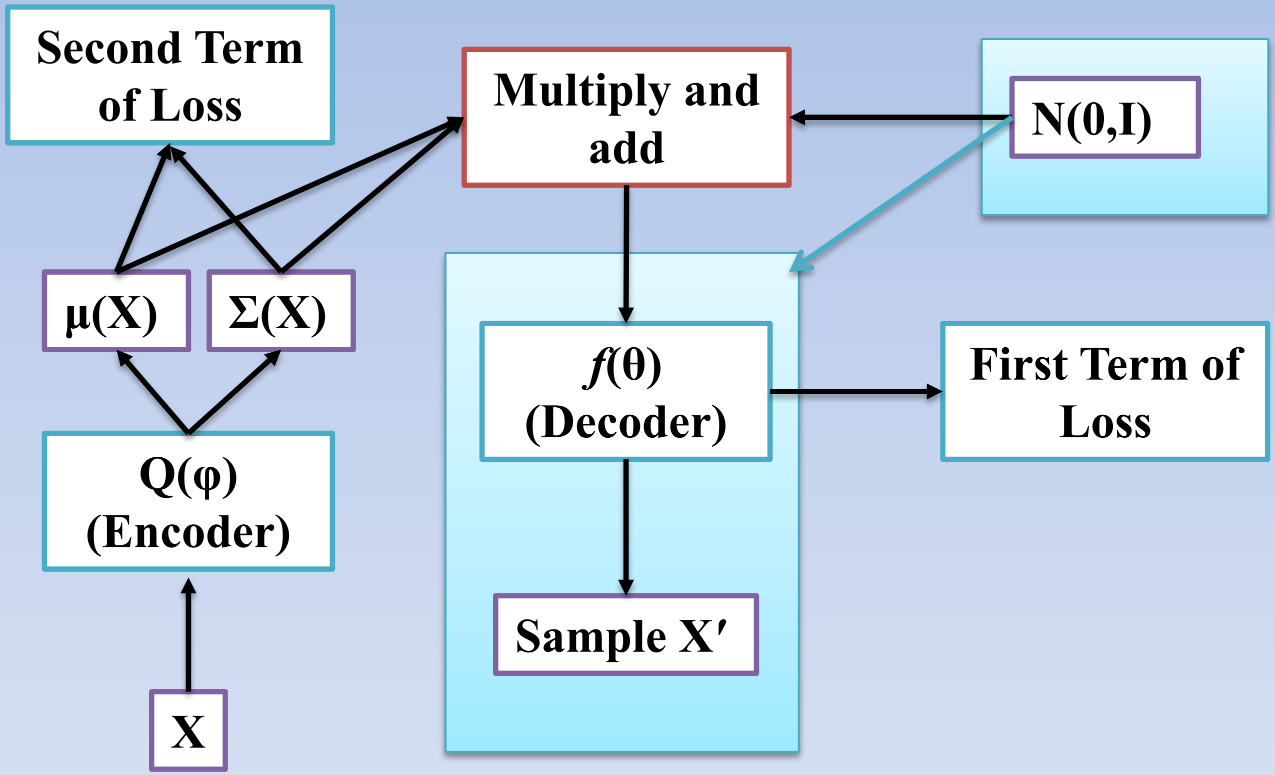
Wait a minute…

Eval phase

The final clever idea: the "Reparameterization Trick"

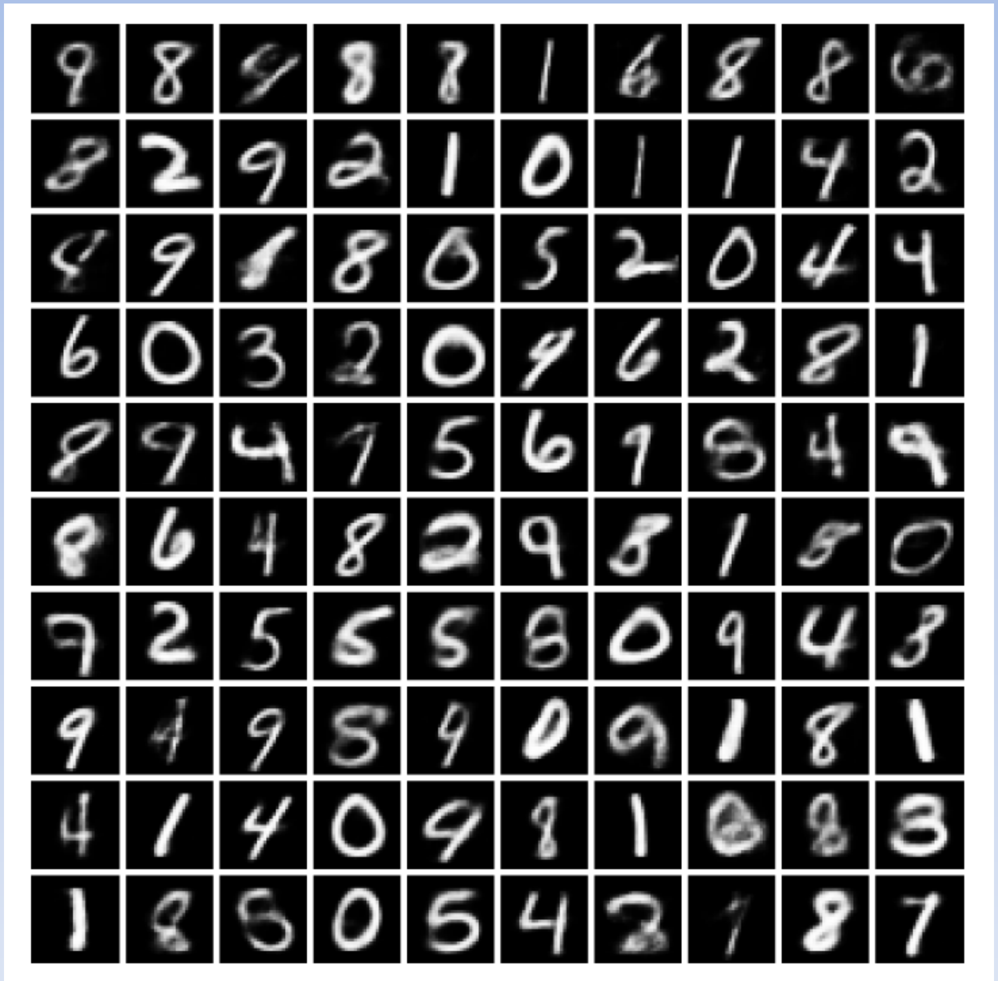
- We cannot backpropagate the loss through the single sample z!
  - So the first term never affects the encoder, which will never learn good choices for z for each X
- So instead, *we move the sampling to the input layer by sampling* $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I})$
- We can do this because for a Gaussian $N(\mu(X), \Sigma(X)) = \mu(X) + \Sigma^{1/2}(X)\boldsymbol{\varepsilon}$

The Variational Auto-Encoder (VAE) architecture



Eval phase

Example Output: MNIST



Improvements

- Many subsequent modifications
  - Conditional VAE, to condition the VAE on known evidence/labels
  - Generative Adversarial Networks (GANs)
    - Combine a generative model with a “discriminator” to enable very high dimensional sampling
    - Many interesting questions emerge, see Robbie Dozier’s 2022 MS Thesis
  - Diffusion Models
    - Producing a single Gaussian distribution over X in one step can be hard
    - What if we did in multiple steps, each step a *perturbation* of the previous?