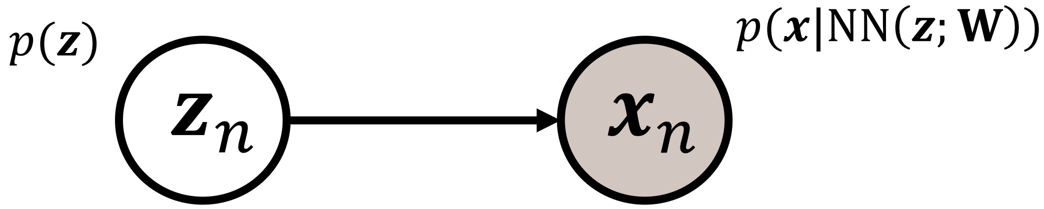
(Deep) Generative Models for Unsupervised Learning (Part 3 – VAE and Amortized Inference)

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Constructing Generative Models using Neural Nets

- We can use a neural net to define the mapping from a K-dim $\boldsymbol{z}_n$ to D-dim $\boldsymbol{x}_n$



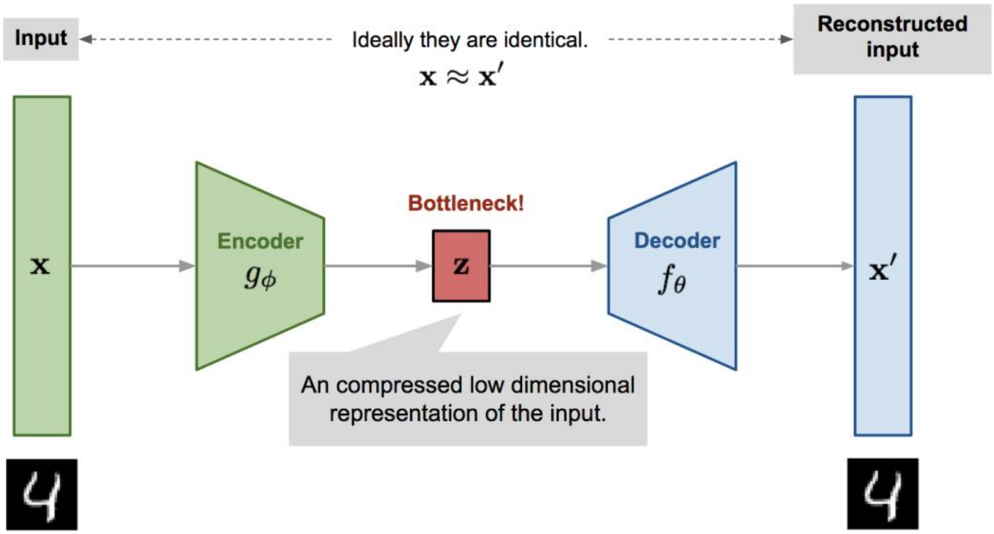
$$\boldsymbol{x}_n \sim \mathcal{N}(\mathrm{NN}(\boldsymbol{z}_n;\mathbf{W}), \sigma^2\mathbf{I}_D)$$
$$\boldsymbol{x}_n \sim \mathrm{ExpFam}(\mathrm{NN}(\boldsymbol{z}_n;\mathbf{W}))$$

- If $\boldsymbol{z}_n$ has a Gaussian prior, such models are called deep linear Gaussian models (DLGM)
- Since NN mapping can be very powerful, DLGM can generate very high-quality data
  - Take the trained network, generate a random $\boldsymbol{z}$ from prior, pass it through the model to generate $\boldsymbol{x}$

Some sample images generated by Vector Quantized Variational AutoEncoder (VQ-VAE), a state-of-the-art DLGM

Variational Autoencoder (VAE)

- VAE* is a probabilistic extension of autoencoders (AE)



$$L_{\mathrm{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

- The basic difference is that VAE assumes a prior $\boldsymbol{p(z)}$ on the latent code $\boldsymbol{z}$
  - This enables it to not just compress the data but also generate <u>synthetic data</u>
  - How: Sample $\boldsymbol{z}$ from a prior and pass it through the decoder
- Thus VAE can learn good latent representation + generate novel synthetic data
- The name has "Variational" in it since it is learned using VI principles

*Autoencoding Variational Bayes (Kingma and Welling, 2013)

Pic source: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Variational Autoencoder (VAE)

- VAE has three main components
  - A prior $p_\theta(z)$ over latent codes
  - A probabilistic decoder $p_\theta(x|z)$
  - A posterior or probabilistic encoder $p_\theta(z|x)$ approx. by an "inference network" $q_\phi(z|x)$

Here θ collectively denotes all the parameters of the prior and likelihood

Using the idea of "Amortized Inference" (next slide)

Here ϕ collectively denotes all the parameters that define the inference network

- VAE is learned by maximizing the ELBO

ELBO for a single data point

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

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \mathbb{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z})\right)$$

Maximized to find the optimal θ and ϕ

q_ϕ should reconstruct the data x well from z (high log-lik)

q_ϕ should also be simple (close to the prior)

- The Reparametrization Trick is commonly used to optimize the ELBO
- Posterior is inferred only over z, and usually only point estimate on θ and ϕ

Amortized Inference

- Latent variable models need to infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ for each observation $\boldsymbol{x}_n$

- This can be slow if we have lots of observations because
  1. We need to iterate over each $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$
  2. Learning the global parameters needs wait for step 1 to finish for all observations

- One way to address this is via Stochastic VI (already saw)

- Amortized inference is another appealing alternative (used in VAE and other LVMs too)

$$p(\boldsymbol{z}_n|\boldsymbol{x}_n) \approx q(\boldsymbol{z}_n|\boldsymbol{\phi}_n) = q(\boldsymbol{z}_n|\mathrm{NN}(\boldsymbol{x}_n;\boldsymbol{W}))$$

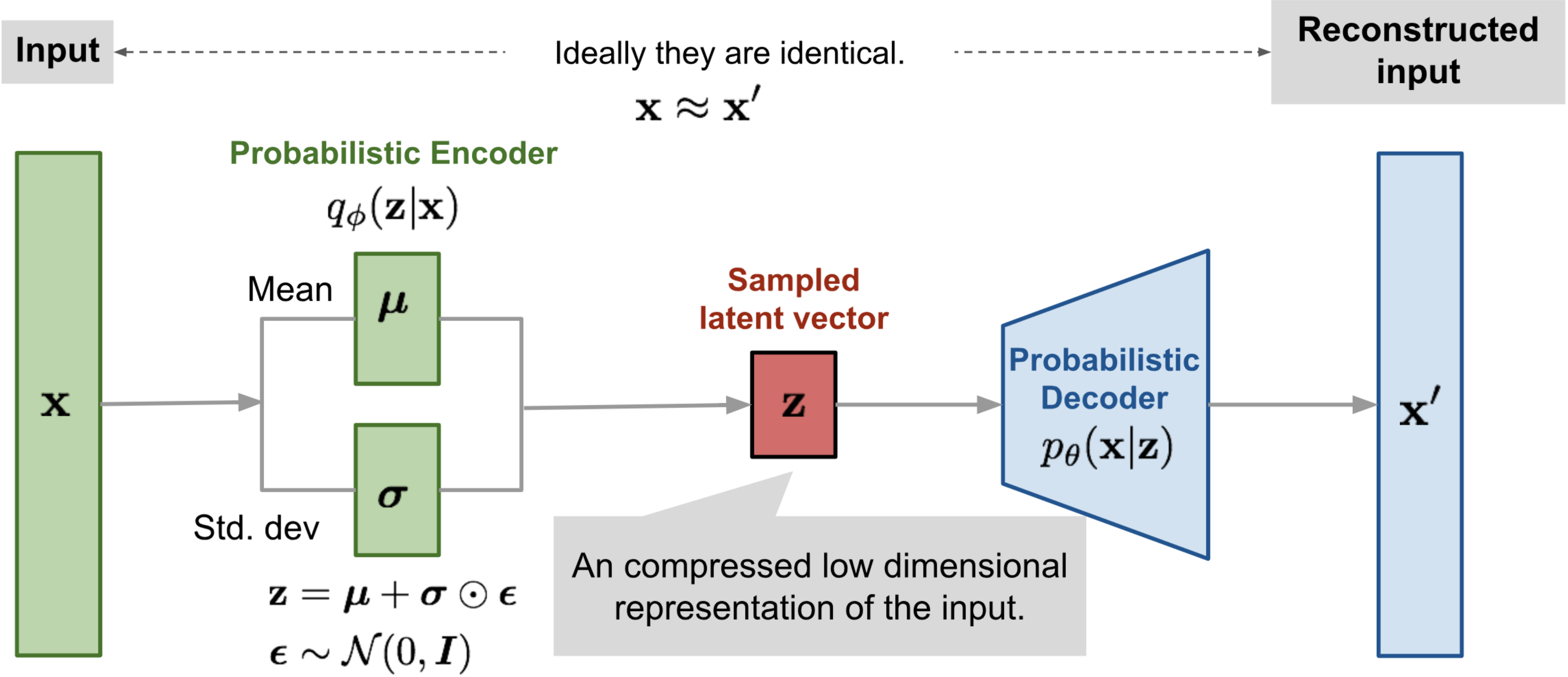
If $\boldsymbol{q}$ is Gaussian then the NN will output a mean and a variance

- Thus no need to learn $\boldsymbol{\phi}_n$'s (one per data point) but just a single NN with params $\boldsymbol{W}$
  - This will be our "encoder network" for learning $\boldsymbol{z}_n$
  - Also very efficient to get $p(\boldsymbol{z}_*|\boldsymbol{x}_*)$ for a new data point $\boldsymbol{x}_*$

Variational Autoencoder: The Complete Pipeline

- Both probabilistic encoder and decoder learned jointly by maximizing the ELBO

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



Pic source: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

VAE and Posterior Collapse

Decoder is a neural net and can be arbitrarily powerful making this term very large

Consequently, KL will become close to zero collapsing posterior to the prior

- VAEs may suffer from posterior collapse

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

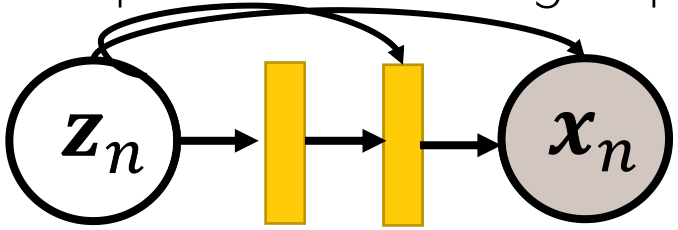
- Thus, due to posterior collapse, reconstruction will still be good but the code $\boldsymbol{z}$ may be garbage (not useful as a representation for $\boldsymbol{x}$)
- Several ways to prevent posterior collapse, e.g.,
  - Use KL annealing

A carefully tuned value between 0 and 1

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

For example, keep the variance of $\boldsymbol{q}$ as fixed

  - Avoid KL from becoming 0 using some $\boldsymbol{q}$ doesn't collapse to the prior
  - More tightly couple $\boldsymbol{z}$ with $\boldsymbol{x}$ using skip-connections (Skip-VAE)

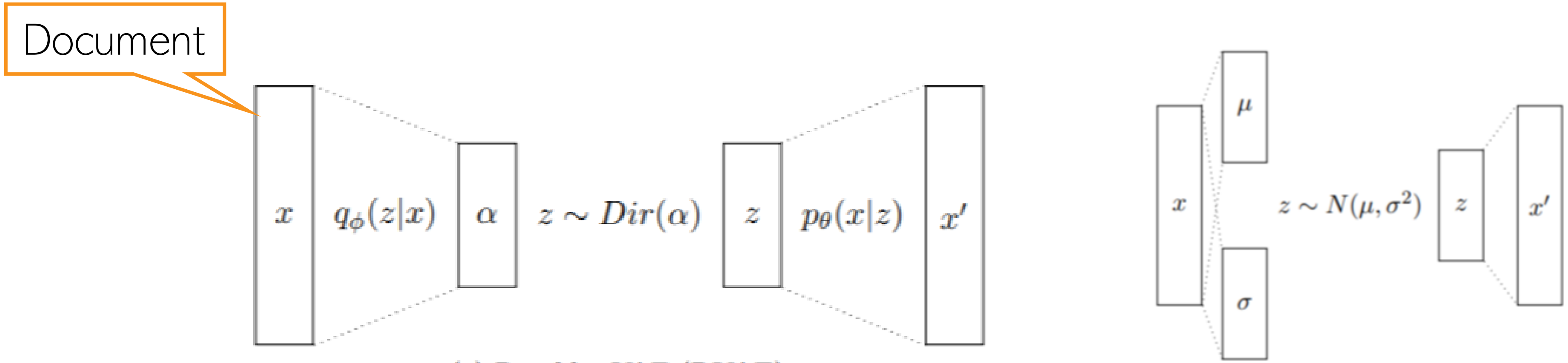


Hidden layers of NN

Besides these, MCMC (sometimes used for inference in VAE), or improved VI techniques can also help in preventing posterior collapse in VAEs

VAE: Some Comments

- One of the state-of-the-art latent variable models
- Useful for both generation as well as representation learning
- Many improvements and extensions, e.g.,
  - For text data and sequences (VAE for topic models or "neural topic models")



(a) Dirichlet VAE (DVAE)

(b) Gaussian VAE (NVDM)

  - Combination of VAE with other deep generative models such as GANs
  - VAE-style models with more than one layer of latent variables (Sigmoid Belief Networks, hierarchical VAE, Ladder VAE, Deep Exponential Families, etc)

Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model (Burkhardt and Kramer, 2020)