Variational Autoencoders

Spring 2020
11-785 Introduction to Deep Learning

By Lady Yuying Zhu
And Lord Christopher George

Outline

- **Autoencoders**
- Bake a cake 🤔 TBD, vanilla please
- *Probability Refresher*
- **Generative Models (basic overview)**
- Take over the universe
- Discuss our eventual doomed existence and the impact of covid-19 on society
- **Variational Autoencoders**

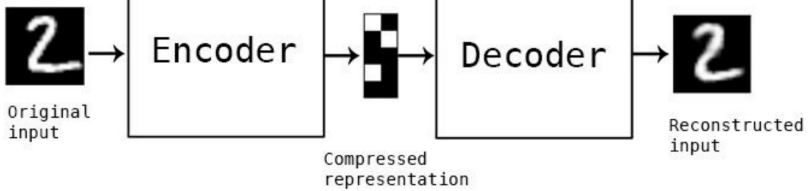
Auto-encoder Recap

A neural network that the output is the input itself.

Intuition:

- A good representation should keep the information well (reconstruction error)
- Deep + nonlinearity might help enhance the representation power

$$\min_{W_1, W_2} \|x_i - g(f(x_i; W_i); W_2)\|_2^2$$



Slide Credits: Shenlong Wang, Deep Generative Models

Why auto-encoder? Why not?

Why?

- Map high-dimensional data to low-dimensions for visualization
- Learn the salient features of the data.

Why not?

Encoded data can be decoded without loss if the autoencoder has enough degrees of freedom (leads to serve overfitting)

Without explicit regularisation, some points of the latent space are “meaningless” once decoded

Irregular latent space prevent us from using autoencoder for new content generation

Probability Refresher

- Continuous vs. Discrete
- Bayes Rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Prior/ Posterior

$$P(A|B) \propto P(B|A)P(A)$$

posterior likelihood prior

- KL Divergence : measure the closeness of the two distributions

$$KL(q||p) = \mathbb{E}_{q(\mathrm{z};\lambda)}[\log q(\mathrm{z};\lambda) - \log p(\mathrm{z}|\mathrm{x})]$$

- ELBO (evidence lower bound)

$$\mathrm{ELBO}(\lambda) = \mathbb{E}_{q(\mathbf{z}\,;\,\lambda)}\left[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}\,;\,\lambda)\right]$$

Generative Models, An Overview

Task: generate new samples follows the same probabilistic distribution of a given a training dataset

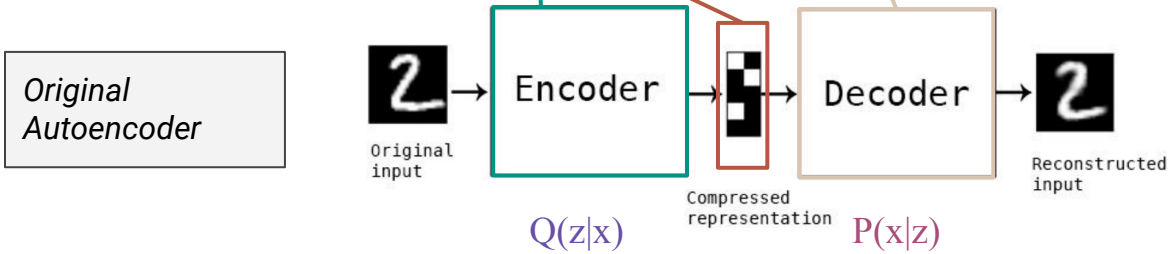
Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Variational Autoencoders from Autoencoders

"In VAEs, the encoder becomes a variational inference network, mapping observed inputs to (approximate) posterior distributions over latent space, and the decoder becomes a generative network, capable of mapping arbitrary latent coordinates back to distributions over the original data space."



Fast Forward Labs, Under the Hood of the Variational Autoencoder (in Prose and Code)

Variational Autoencoders, (VAEs)

Big Idea: Maximize likelihood of seeing our data. “We are aiming maximize the probability of each X in the training set under the entire generative process.”

$$P(X) = \int P(X|z;\theta)P(z)dz$$

Where X is a datapoint, z is the latent variable, and theta are the parameters used for calculating z.

Generally in VAEs $P(X|z;\theta) = \mathcal{N}(X|f(z;\theta), \sigma^2 * I)$

And $P(z) = \mathcal{N}(z|0, I)$

What could this be?

Note: Why are we doing this? Because Variational Autoencoders can be interpreted as using variational bayesian inference where, in this bayesian view, our data is seen as being pulled from some underlying distribution. x ~ p(x)

Tutorial on Variational Autoencoders, Section 1.1 Preliminaries: Latent Variable Models
Fast Forward Labs, Introducing Variational Autoencoders (in Prose and Code)

VAEs, Question 1

How do we define the latent variables z (i.e., decide what information they represent)?

* Samples of z can be drawn from a simple distribution, i.e., N (0, I), where I is the identity matrix.

* Provided powerful function approximators, we can simply learn a function which maps our independent, normally-distributed z values to whatever latent variables might be needed for the model, and then map those latent variables to X.

$$P(X|z;\theta) = \mathcal{N}(X|f(z;\theta), \sigma^2 * I)$$

A Neural Network!

Tutorial on Variational Autoencoders, Section 2, Variational Autoencoders

VAEs, Question 2

How do we deal with the integral over z?

* VAEs alter the sampling procedure to make it faster!

* The key idea behind the variational autoencoder is to attempt to sample values of z that are likely to have produced X, and compute P(X) just from those. This means that we need a new function Q(z|X) which can take a value of X and give us a distribution over z values that are likely to produce X.

* Given this Q(Z|x), we can compute P(X|z) relatively easily, so…

Tutorial on Variational Autoencoders, Section 2, Variational Autoencoders

VAEs and KL Divergence

So now we need to relate E_{z~Q}P(X|z) and P(X), which we can do by minimizing the KL Divergence:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$ By definition

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$ Apply Bayes Rule

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$ Rearrange

$$\log P(X) - \mathcal{D}\left[Q(z|X)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]$$ Reconstruct Q

$$\log P(X|z) - \mathcal{D}\left[Q(z|X)\|P(z)\right]$$ Take the gradient of RHS

Minimizing KL Divergence = Maximizing ELBO

Tutorial on Variational Autoencoders, Section 2, Variational Autoencoders

VAEs and KL Divergence

$$\log P(X|z) - \mathcal{D}\left[Q(z|X) \| P(z)\right]$$

Minimizing KL Divergence = Maximizing ELBO

Negative Reconstruction Loss: Can take sample of z and treat P(X|z) as an approximation, then backpropagate through the neural network. → Reparameterization trick.

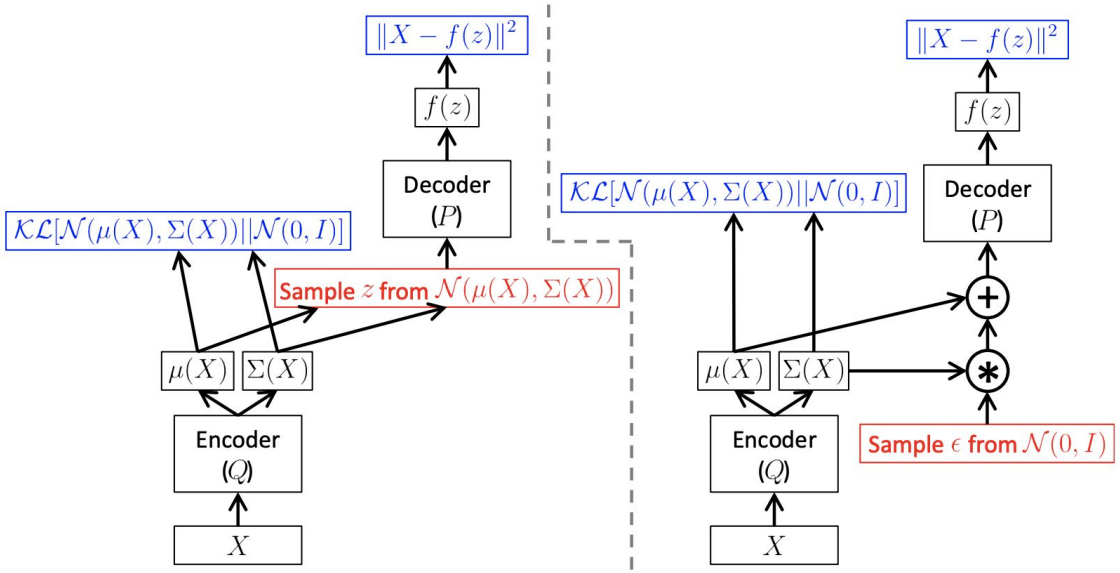
KLD Loss: Has a closed form solution

$$\mathcal{D}[\mathcal{N}(\mu(X), \Sigma(X)) \| \mathcal{N}(0, I)] = \frac{1}{2}\left(\text{tr}\left(\Sigma(X)\right) + \left(\mu(X)\right)^{\top}\left(\mu(X)\right) - k - \log\det\left(\Sigma(X)\right)\right)$$

Note: ELBO is the Evidence Lower Bound, and can be reformulated as the equation above.

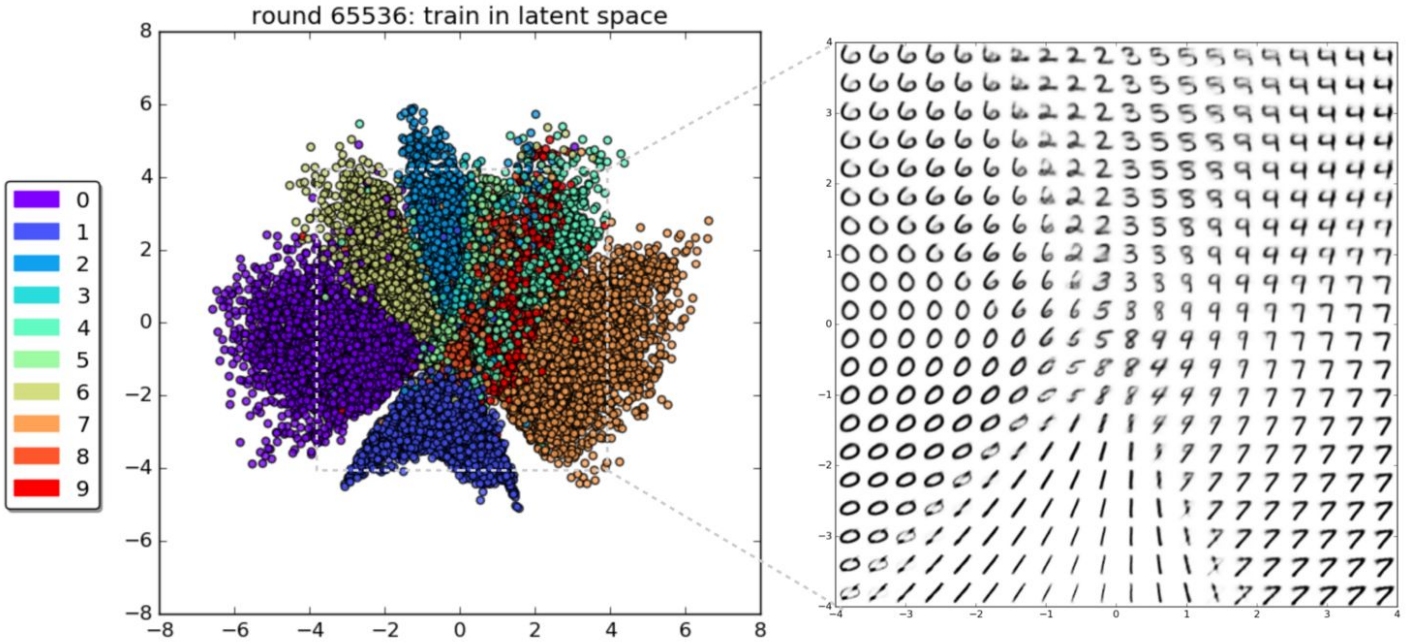
Tutorial on Variational Autoencoders, Section 2, Variational Autoencoders

VAEs, Reparameterization



Tutorial on Variational Autoencoders, Section 2.2 Optimizing the objective
Deep Generative Models, Section 20.9 Back-Propagation through Random Operations

VAEs, Example



Fast Forward Labs, Introducing Variational Autoencoders (in Prose and Code)

References

(1) Shenlong, Wang Deep Generative Models

(2) Chapter 20, Deep Generative Models

(3) Tutorial on Variational Autoencoders

(4) Fast Forward Labs, Under the Hood of the Variational Autoencoder (in Prose and Code)

(5) Fast Forward Labs, Introducing Variational Autoencoders (in Prose and Code)

(6) examples/main.py at master · pytorch/examples

(7) CS231n, stanford