Variational Autoencoders (VAE)

Prashant Shekhar

March 5, 2023

Table of Contents

Generative modeling example

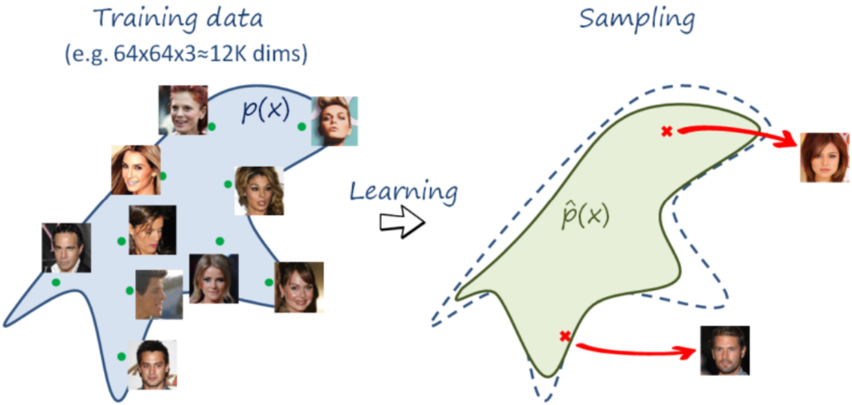


Figure 1: Generative modeling and sampling

Image credits: `http://www.lherranz.org/2018/08/07/imagetranslation/`

Motivation for generative modeling

- **Generative model as a discriminator**: For instance, we have a generative model for an earthquake of type A and another for type B, then seeing which of the two describes the data best we can compute a probability for whether earthquake A or B happened.
- **Generative models to assist classifiers**: For instance, one may have few labeled examples and many more unlabeled examples. In this semi-supervised learning setting, one can use the generative model of the data to improve classification.
- **Generative model as a regularizer**: By forcing the representations/generative model to be as a meaningful as possible, we bias the inverse of that process, which maps from input to representation, into a certain mould.

A typical VAE

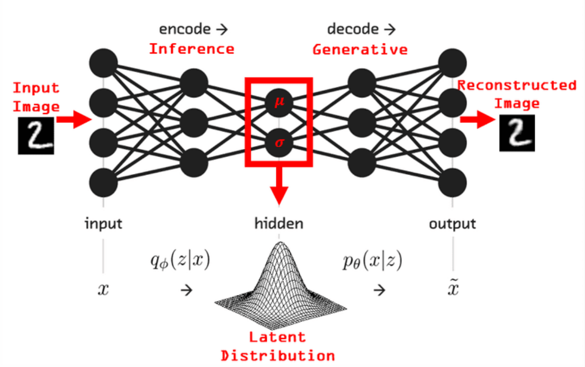


Figure 2: A typical VAE for synthesizing handwritten digits. The VAE can be viewed as two coupled, but independently parameterized models: the **encoder/inference/recognition** model, and the **decoder/generative model**. These two models support each other and are jointly optimized.

Image credits: `https://theaisummer.com/Autoencoder/`

The problem solved by VAEs

- We often collect dataset D consisting of $n \geq 1$ samples:

$$D = \{x_1, x_2, ..., x_n\} \equiv \{x_i\}_{i=1}^n$$

  these samples x_i are independent and identically distributed (i.i.d)
- We assume the observed samples x_i are random samples from an unknown underlying process, whose true (probability) distribution $p^*(x)$ is unknown.
- We attempt to approximate this underlying process with a chosen model $p_\theta(x)$ with parameters θ such that:

$$x_i \sim p_\theta(x)$$

- Hence, training a VAE is equivalent to find the best value of θ such that for any observed sample x_i

$$p_\theta(x_i) \approx p^*(x_i)$$

- Once you have found such a θ, you can use $p_\theta(x)$ to even draw a new sample x_j which was not a part of the training set used to fit the VAE.

Whats a latent variable ?

- Latent variables are variables that are part of the model, but which we don't observe, and are therefore not part of the dataset D. We typically use z to denote such latent variables.
- For VAEs or autoencoders, z represents the underlying 'simpler' latent representations that map to samples x. This relationship prescribes a joint distribution over x and z: $p(x, z)$. We need z to account for complicated things that might occur in this world.
- Hence the distribution which VAE is trying to learn ($p_\theta(x)$) is a marginal distribution:

$$p_\theta(x) = \int p_\theta(x, z)dz \tag{1}$$

$p_\theta(x)$ is also referred to as *(single datapoint) marginal likelihood*.

Marginal likelihood

- Because of the i.i.d assumption the *marginal likelihood* of the dataset D is given as:

$$p_\theta(D) = \prod_{i=1}^{n} p_\theta(x_i) \tag{2}$$

or the log marginal likelihood

$$log\ \ p_\theta(D) = \sum_{i=1}^{n} log\ \ p_\theta(x_i) \tag{3}$$

- However, we dont have an efficient estimator for $p_\theta(x) = \int p_\theta(x, z)dz$. Even with the below mentioned **monte carlo estimate**, we will potentially need a lot of z samples to approximate $p_\theta(x)$:

$$p_\theta(x) = \frac{1}{m} \sum_{i=1}^{m} p_\theta(x|z^m)$$

hence we cannot compute or directly optimize the log-marginal likelihood (3) for optimizing the parameters θ. **Hence the log-marginal likelihood is intractable**.

Dealing with Intractability

- Source of intractability (can't be accurately computed):

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

  - $p_\theta(z|x)$: Intractable
  - $p_\theta(x,z)$: Tractable
  - $p_\theta(x)$: Intractable

  Hence the intractability of $p_\theta(z|x)$ and $p_\theta(x)$ are related to each other.
- Approximate inference techniques will allow us to approximate the posterior $p_\theta(z|x)$. For this, we introduce a parametric inference model $q_\phi(z|x)$ and **optimize** ϕ **such that**:

$$q_\phi(z|x) \approx p_\theta(z|x)$$

- This also helps us optimize marginal likelihood $p_\theta(x)$ to get the best parameters θ.
- From now we will call θ as **model parameters** and ϕ as **variational parameters**.

Overall picture till now: VAE

- A VAE learns stochastic mappings between an observed $x - space$, whose empirical distribution is typically complicated, and a latent $z - space$, whose distribution can be relatively simple (such as spherical, as in this figure).
- The generative model learns a joint distribution $p_\theta(x, z)$ that is often (but not always) factorized as $p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$, with a prior distribution over latent space $p_\theta(z)$, and a stochastic decoder $p_\theta(x|z)$.
- The stochastic encoder $q_\phi(z|x)$, also called inference model, approximates the true but intractable posterior $p_\theta(z|x)$ of the generative model.

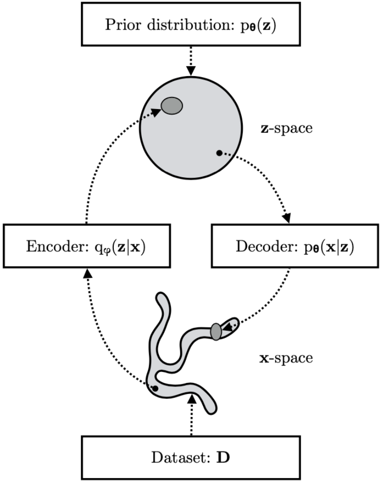
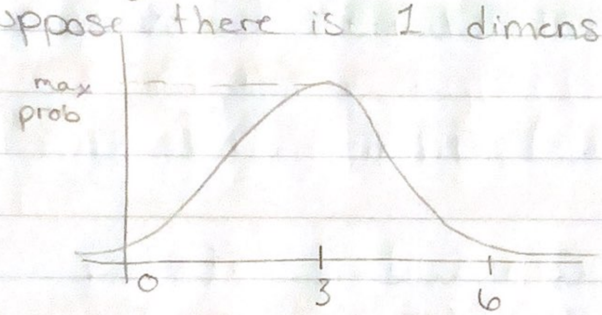


Image credits: `https://arxiv.org/pdf/1906.02691.pdf`

03/02/2023 Variational Autoencoders (VAE)

- generative modeling example
  - $P(x) = N(\mu, WW^T + \sigma^2 I)$
  - generating a new sample by learning $P(x)$ and sampling from $P(x)$
  - suppose there is 1 dimension



$\mu = 3 \quad \sigma^2 = 1$

$x \sim N(\mu, \sigma^2)$

sample from here

- objective is to learn a distribution
- motive for generative modeling
  - could be used as a discriminator, to assist classifiers, and as a regularizer
  - discriminator
    - which probability dist. does the new sample follow the most?
  - assist
    - could improve classification
  - regularizer
    - generalizes better by restricting it to follow $z \sim N(\mu, \Sigma)$
- typical VAE
  - encoder = inference $q_\phi(z|x)$
  - decoder = generative $p_\theta(x|z)$

- encoder and decoder are jointly optimized (ϕ and θ)
- problem solved by VAE
  - all samples x_i are independent and identically distributed (pdf is unknown, but same)
  - $p^*(x)$ is true prob dist
  - $x_i \sim p_\theta(x_i)$
  - VAE tries to make $p_\theta(x_i)$ equivalent (as close as possible) $p_\theta(x_i) \approx p^*(x_i)$ then draws new sample x_j from $p_\theta(x)$
- latent variable
  - z_i are independent variables that arguably exist in a simpler representation
  - not observed
  - only observe x
  - z accounts for noise
  - $p(x, z) = p(x \mid z)p(z)$
- marginal likelihood
  - no efficient estimator for $p_\theta(x) = \int p_\theta(x \mid z)p(z)dz$ (can't do integral)
  - monte carlo estimate

$$p_\theta(x) = \frac{1}{m} \sum_{i=1}^{m} p_\theta(x \mid z^m)$$

- monte carlo is estimation by simulation

$$\lim_{m \to \infty} \sum_{i=1}^{m} p_\theta(x|z^m) \frac{1}{m}$$ → approx. of $p(z)$

  - finite term estimate of $\int P(x,z)\,dz$
  - $p_\theta(x|z^m)$ = neural network $(z) \to x$
  - $p(z)$ is a hyperparameter!!!

- dealing with intractability (can't be accurately computed)

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

$p_\theta(z|x)$: intractable; $p_\theta(x,z)$ ← tractable; $p_\theta(x)$: intractable

  - posterior $p_\theta(z|x)$ inverse of what you're observing; updated info about z
  - no data $p(z)$ is based on prior
  - infinite data $p(z)$ is based on data
  - update prior each time

$$q_\phi(z|x) \approx p_\theta(z|x)$$

$$p^*(x) \approx p_\theta(x)\{p_\theta(z|x)\} \approx q_\phi(z|x)$$

  - θ is model parameters
  - ϕ is variational parameters