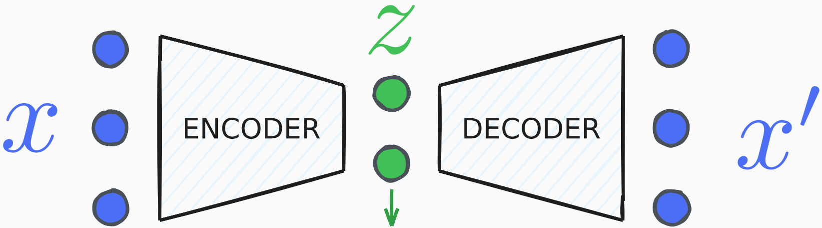
An introduction to generative models

Thomas Gerald

April 11, 2024

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS

Auto-Encoder

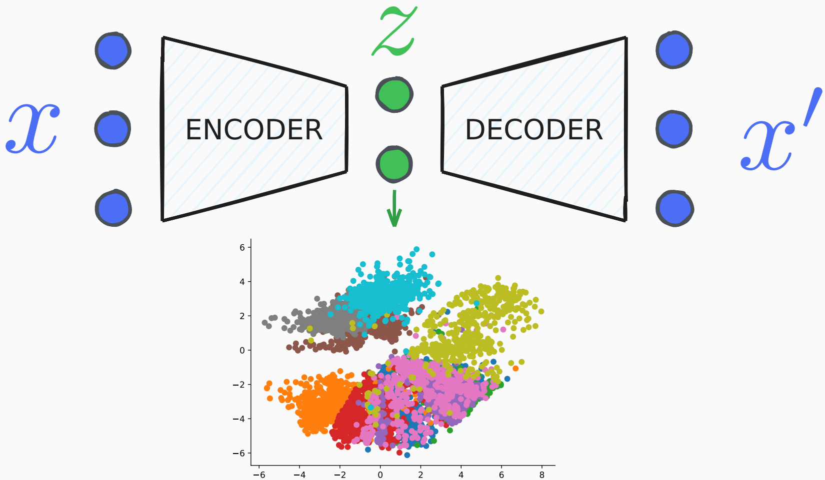


Auto-encoder network
An AutoEncoder is a two function model with

- f encode information
- g decode information

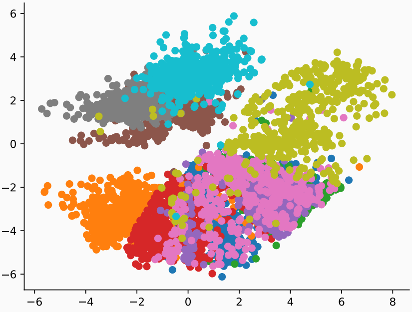
Can we generate from the latent space?

Auto-Encoder



Can we generate from the latent space?

→ **Estimate the distribution of z and sample from this distribution !!**



Estimate latent distribution

- What is $p(z)$?
- What prior?

Difficult to choose the probability function (Gaussian ?)

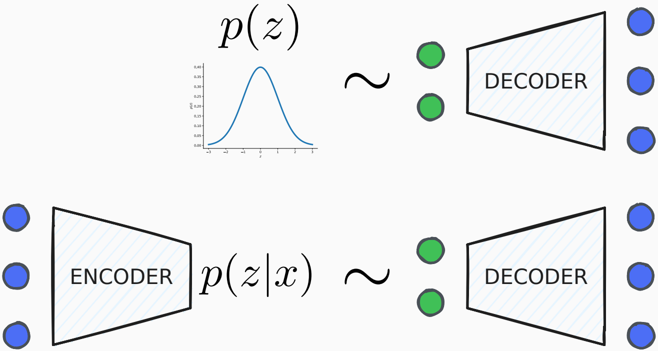
- Ensuring latent representation follow a prior distribution?
- Using a regularisation approach to force data follow a density function?

Variational Auto-Encoder

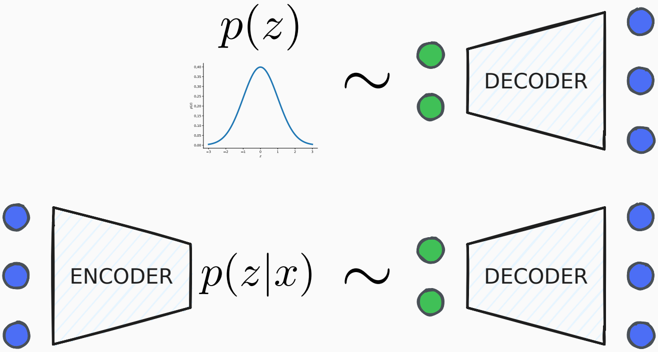
- Ensuring latent representation follow a prior distribution?
- Using a regularisation approach to force data follow a density function?

$\rightarrow$ It is the variational encoder intuition!!!

Variational Auto-Encoder



Variational Auto-Encoder



The variational Auto-Encoder (main idea)

- Force latent representation to fit a distribution
- Decode samples to produce data

Estimate $p(z|x)$ knowing $p(z)$

The variational Auto-Encoder

Let consider $p(z|x)$ the conditionnal probability of sampling a latent representation prior to x.

Using Baye's rule we have:

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

- $p(z|x)$ the posterior probability $\rightarrow$ we want to estimate

The variational Auto-Encoder

Let consider $p(z|x)$ the conditionnal probability of sampling a latent representation prior to x.

Using Baye's rule we have:

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

- $p(z|x)$ the posterior probability $\rightarrow$ we want to estimate
- $p(z)$ the prior $\rightarrow$ we choose one (for instance gaussian)

The variational Auto-Encoder

Let consider $p(z|x)$ the conditionnal probability of sampling a latent representation prior to x.

Using Baye's rule we have:

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

- $p(z|x)$ the posterior probability $\rightarrow$ we want to estimate
- $p(z)$ the prior $\rightarrow$ we choose one (for instance gaussian)
- $p(x|z)$ the likelihood $\rightarrow$ approximable (reconstruction)

The variational Auto-Encoder

Let consider $p(z|x)$ the conditionnal probability of sampling a latent representation prior to x.

Using Baye's rule we have:

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

- $p(z|x)$ the posterior probability $\rightarrow$ we want to estimate
- $p(z)$ the prior $\rightarrow$ we choose one (for instance gaussian)
- $p(x|z)$ the likelihood $\rightarrow$ approximable (reconstruction)
- $p(x)$ the marginal probability/or evidence probability

Variational Auto-Encoder

Denominator $p(x) = \int_z p(x,z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

Variational Auto-Encoder

Denominator $p(x) = \int_z p(x,z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

How to handle it?

- Approximate using Monte-Carlo estimation (sampling to estimate distribution)
- **Variational inference**: choose an other distribution $q(z|x)$ to approximate $p(z|x)$

Denominator $p(x) = \int_z p(x,z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

How to handle it?

- Approximate using Monte-Carlo estimation (sampling to estimate distribution)
- **Variational inference**: choose an other distribution $q(z|x)$ to approximate $p(z|x)$

Variational Inference

Variational Auto-Encoder

Denominator $p(x) = \int_z p(x, z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

How to handle it?

- Approximate using Monte-Carlo estimation (sampling to estimate distribution)
- **Variational inference**: choose an other distribution $q(z|x)$ to approximate $p(z|x)$

Variational Inference

- Let consider $q_\phi(z|x)$ parametrized by ϕ (a neural network)

Variational Auto-Encoder

Denominator $p(x) = \int_z p(x,z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

How to handle it?

- Approximate using Monte-Carlo estimation (sampling to estimate distribution)
- **Variational inference**: choose an other distribution $q(z|x)$ to approximate $p(z|x)$

Variational Inference

- Let consider $q_\phi(z|x)$ parametrized by ϕ (a neural network)
- We would like $q_\phi(z|x)$ close to $p(z|x)$

Variational Auto-Encoder

Denominator $p(x) = \int_z p(x,z)dz = \int_z p(x|z).p(z)dz$ is computationnaly untractable.

How to handle it?

- Approximate using Monte-Carlo estimation (sampling to estimate distribution)
- **Variational inference**: choose an other distribution $q(z|x)$ to approximate $p(z|x)$

Variational Inference

- Let consider $q_\phi(z|x)$ parametrized by ϕ (a neural network)
- We would like $q_\phi(z|x)$ close to $p(z|x)$

$\rightarrow$ Minimize a distance between $q_\phi(z|x)$ and $p(z|x)$

Kullback Leibler divergence

The KL-divergence is a metric between two distribution where the KL-divergence of two distribution is given by:

$$KL\left(p(x)||q(x)\right) = \int_x p(x)\log\frac{p(x)}{q(x)}$$

- When distributions are the same $KL\left(p(x)||q(x)\right) = 0$
- When distributions are not the same $0 < KL\left(p(x)||q(x)\right)$

F

In our case we want to **minimize** $KL\left(q(z|x)||p(z|x)\right)$

Kullback Leibler divergence

The KL-divergence is a metric between two distribution where the KL-divergence of two distribution is given by:

$$KL\left(p(x)||q(x)\right) = \int_x p(x) \log \frac{p(x)}{q(x)}$$

- When distributions are the same $KL\left(p(x)||q(x)\right) = 0$
- When distributions are not the same $0 < KL\left(p(x)||q(x)\right)$

F

In our case we want to **minimize** $KL\left(q(z|x)||p(z|x)\right)$

⚠The Kullback Leibler divergence is not symetric !!!

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$KL\left(q(z|x)||p(z|x)\right)$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)}
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right]
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x)
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)}
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)} \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \int_z q(z|x) \log p(x)
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)} \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \int_z q(z|x) \log p(x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x) \int_z q(z|x)
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)} \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \int_z q(z|x) \log p(x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x) \int_z q(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x)
\end{aligned}
$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)} \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \int_z q(z|x) \log p(x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x) \int_z q(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x)
\end{aligned}
$$

In rearanging terms we get we can have the expression of the evidence:

$$\log p(x) = KL\left(q(z|x)||p(z|x)\right) - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] + \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right]$$

Variational Auto-Encoder: $KL\left(q(z|x)||p(z|x)\right)$

$$
\begin{aligned}
& KL\left(q(z|x)||p(z|x)\right) \\
= \quad & \int_z q(z|x) \log \frac{q(z|x)}{p(z|x)} \\
= \quad & \int_z q(z|x) \left[\log q(z|x) - \log(p(z|x))\right] \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log \frac{p(z,x)}{p(x)} \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \int_z q(z|x) \log p(x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x) \int_z q(z|x) \\
= \quad & \int_z q(z|x) \log q(z|x) - \int_z q(z|x) \log p(z,x) + \log p(x)
\end{aligned}
$$

In rearanging terms we get we can have the expression of the evidence:

$$\log p(x) = KL\left(q(z|x)||p(z|x)\right) - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] + \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right]$$

Yes but... we don't know (or can estimate) $p(z|x)$

Variational Auto-Encoder: The evidence lower bound

$$\log p(x) = KL\left(q(z|x)||p(z|x)\right) - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] + \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right]$$

$KL\left(q(z|x)||p(z|x)\right)$ is intractable however:

$\rightarrow$ The Kullback Leibler divergence is positive

Variational Auto-Encoder: The evidence lower bound

$$\log p(x) = KL\left(q(z|x)||p(z|x)\right) - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] + \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right]$$

$KL\left(q(z|x)||p(z|x)\right)$ is intractable however:

$\rightarrow$ The Kullback Leibler divergence is positive

Thus $\log p(x) \geq -\mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] + \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right]$

$$\text{Maximize } \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right]$$

$$\Downarrow$$

$$\text{Maximize } \log p(x)$$

Variational Auto-Encoder objective

Maximize $\mathbb{E}_{z\sim q(z|x)}[\log p(z,x)] - \mathbb{E}_{z\sim q(z|x)}[\log q(z|x)]$

This lower bound is called the Evidence Lower Bound (**ELBO**)

Variational Auto-Encoder: The evidence lower bound

Variational Auto-Encoder objective

Maximize $\mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right]$

This lower bound is called the Evidence Lower Bound (**ELBO**)

- $p(x)$ is often call the evidence (probability of input data)
- $\log p(x) \leq p(x)$
- The ELBO is a lower bound of the evidence

Variational Auto-Encoder: The evidence lower bound

Variational Auto-Encoder objective

Maximize $\mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right]$

This lower bound is called the Evidence Lower Bound (**ELBO**)

- $p(x)$ is often call the evidence (probability of input data)
- $\log p(x) \leq p(x)$
- The ELBO is a lower bound of the evidence

We have not finish yet:

Variational Auto-Encoder: The evidence lower bound

Variational Auto-Encoder objective

Maximize $\mathbb{E}_{z\sim q(z|x)}[\log p(z,x)] - \mathbb{E}_{z\sim q(z|x)}[\log q(z|x)]$

This lower bound is called the Evidence Lower Bound (**ELBO**)

- $p(x)$ is often call the evidence (probability of input data)
- $\log p(x) \leq p(x)$
- The ELBO is a lower bound of the evidence

We have not finish yet:

- How to estimate $\log p(z,x)$?

Variational Auto-Encoder: The evidence lower bound

Variational Auto-Encoder objective

Maximize $\mathbb{E}_{z \sim q(z|x)} [\log p(z, x)] - \mathbb{E}_{z \sim q(z|x)} [\log q(z|x)]$

This lower bound is called the Evidence Lower Bound (**ELBO**)

- $p(x)$ is often call the evidence (probability of input data)
- $\log p(x) \leq p(x)$
- The ELBO is a lower bound of the evidence

We have not finish yet:

- How to estimate $\log p(z, x)$?
- Estimate update of $q_\theta(z|x)$ (especially because we sample on $q_\theta(z|x)$)

Variational Auto-Encoder: The evidence lower bound

$$ELBO = \mathbb{E}_{z \sim q(z|x)} [\log p(z, x)] - \mathbb{E}_{z \sim q(z|x)} [\log q(z|x)]$$

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right]
\end{aligned}
$$

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z \sim q(z|x)}\left[\log p(z, x)\right] - \mathbb{E}_{z \sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z \sim q(z|x)}\left[p(z) \log p(x|z)\right] - \mathbb{E}_{z \sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z \sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z \sim q(z|x)}\left[\log q(z|x) - \log p(z)\right]
\end{aligned}
$$

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right]
\end{aligned}
$$

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - KL\left(q(z|x)||p(z)\right)
\end{aligned}
$$

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - KL\left(q(z|x)||p(z)\right)
\end{aligned}
$$

- We know $q_\phi(z|x) \rightarrow$ **Encoder**
- We know $p(z)$ (The distribution we want to have on the latent space)
- We know $p_\theta(x|z) \rightarrow$ **Decoder**
- We don't know the derivative when sampling $\mathbb{E}_{z\sim q(z|x)}$ (for backpropagation)

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - KL\left(q(z|x)||p(z)\right)
\end{aligned}
$$

We can remarks that we have two terms:

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - KL\left(q(z|x)||p(z)\right)
\end{aligned}
$$

We can remarks that we have two terms:

- $\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] \rightarrow$ fitting the data

Variational Auto-Encoder: The evidence lower bound

$$
\begin{aligned}
ELBO &= \mathbb{E}_{z\sim q(z|x)}\left[\log p(z,x)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[p(z)\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log q(z|x) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{z\sim q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z)}\right)\right] \\
&= \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - KL\left(q(z|x)||p(z)\right)
\end{aligned}
$$

We can remarks that we have two terms:

- $\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] \rightarrow$ fitting the data
- $-KL\left(q(z|x)||p(z)\right) \rightarrow$ for the regularisation of the latent space

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z\sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z\sim q_\phi}[f(x)]$?

Using gradient descent:

- Have $\nabla_\phi \mathbb{E}_{z\sim q_\phi}[f(x)] = \mathbb{E}_{z\sim q_\phi}[f(x)\nabla_\phi \log(q_\phi(z))]$
- Estimate by Monte-Carlo: $\frac{1}{L}\sum_{l=1}^{L}\left[f(x)\nabla_\phi \log(q_\phi(z^l))\right]$ (L samples)

[1] https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using gradient descent:

- Have $\nabla_\phi \mathbb{E}_{z \sim q_\phi}[f(x)] = \mathbb{E}_{z \sim q_\phi}[f(x) \nabla_\phi \log(q_\phi(z))]$
- Estimate by Monte-Carlo: $\frac{1}{L} \sum_{l=1}^{L} \left[f(x) \nabla_\phi \log(q_\phi(z^l)) \right]$ (L samples)

$\rightarrow$ Bad estimator (high variance)
$\rightarrow$ Connexion with Policy Gradient in reinforcement Learning[1]

[1] https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

- Sample some noise $\epsilon \sim p(\epsilon)$

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

- Sample some noise $\epsilon \sim p(\epsilon)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x)$

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

- Sample some noise $\epsilon \sim p(\epsilon)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x)$
- g_ϕ should be differentiable

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}\left[f(x)\right]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}\left[f(x)\right]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

- Sample some noise $\epsilon \sim p(\epsilon)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x)$
- g_ϕ should be differentiable
- Estimate with Monte-Carlo $\mathbb{E}_{z \sim q_{\phi(z|x)}}\left[f(x)\right] \approx \frac{1}{L}\sum_{l=1}^{L}\left[f(g_\phi(\epsilon^l, x))\right]$

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}\left[f(x)\right]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}\left[f(x)\right]$?

Using the reparametrization trick:

Estimate $z \sim q_\phi(z|x)$ by using a differentiable transformation:

- Sample some noise $\epsilon \sim p(\epsilon)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x)$
- g_ϕ should be differentiable
- Estimate with Monte-Carlo $\mathbb{E}_{z \sim q_\phi(z|x)}\left[f(x)\right] \approx \frac{1}{L}\sum_{l=1}^{L}\left[f(g_\phi(\epsilon^l, x))\right]$

Notice that this is the Variational Auto-Encoder approach[2]

[2] **Original Paper:** Auto-Encoding Variational Bayes, Kingma And Welling, 2013

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization Gaussian case:

Estimate $z_i \sim q_\phi(z|x) = \mathcal{N}(\mu_i, \sigma_i^2)$ (in practice the encoder predict μ and σ for latent component):

- Sample some noise $\epsilon \sim \mathcal{N}(0, 1)$

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization Gaussian case:

Estimate $z_i \sim q_\phi(z|x) = \mathcal{N}(\mu_i, \sigma_i^2)$ (in practice the encoder predict μ and σ for latent component):

- Sample some noise $\epsilon \sim \mathcal{N}(0, 1)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x) = \epsilon * \sigma_i + \mu_i$

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z \sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z \sim q_\phi}[f(x)]$?

Using the reparametrization Gaussian case:

Estimate $z_i \sim q_\phi(z|x) = \mathcal{N}(\mu_i, \sigma_i^2)$ (in practice the encoder predict μ and σ for latent component):

- Sample some noise $\epsilon \sim \mathcal{N}(0, 1)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x) = \epsilon * \sigma_i + \mu_i$
- Estimate with Monte-Carlo $\mathbb{E}_{z \sim q_\phi(z|x)}[f(x)] \approx \frac{1}{L} \sum_{l=1}^{L} \left[f(\epsilon^{(l)} * \sigma_i + \mu_i) \right]$

Variational Auto-Encoder: The reparametrization trick

Estimate ϕ for $\mathbb{E}_{z\sim q_\phi}[f(x)]$

Finding parameters ϕ of the distribution considering loss $\mathbb{E}_{z\sim q_\phi}[f(x)]$?

Using the reparametrization Gaussian case:

Estimate $z_i \sim q_\phi(z|x) = \mathcal{N}(\mu_i, \sigma_i^2)$ (in practice the encoder predict μ and σ for latent component):

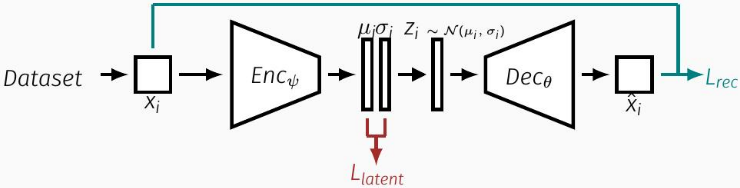
- Sample some noise $\epsilon \sim \mathcal{N}(0,1)$
- Apply the transformation $\hat{z} = g_\phi(\epsilon, x) = \epsilon * \sigma_i + \mu_i$
- Estimate with Monte-Carlo $\mathbb{E}_{z\sim q_\phi(z|x)}[f(x)] \approx \frac{1}{L}\sum_{l=1}^{L}\left[f(\epsilon^{(l)} * \sigma_i + \mu_i)\right]$

In most implementation, only one sample is used in the Monte-Carlo approximation

Variational Auto-Encoder: Summary

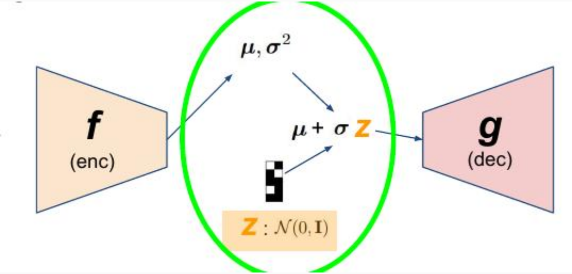
- Latent representation follow a distribution $p(z)$
- $q(z|x)$ encode data (It is a neural network)
- $p(x|z)$ decode data (It is also a neural network)
- We minimize the ELBO loss
- We sample in forward according to a distribution

Variational Auto-Encoder: Framework

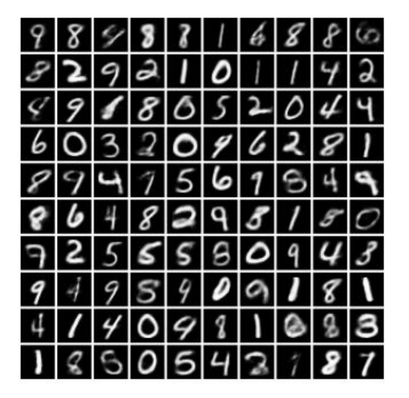


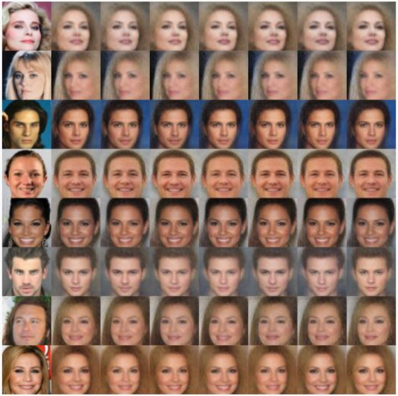
- encoding cost: $L_{latent} = \sum_i D_{KL}\left(Enc_\psi(x_i) \,||\, \mathcal{N}(0;1)\right)$
- reconstruction loss:

$$
\begin{aligned}
L_{rec} &= \sum_i \mathbb{E}_{z \sim Enc_\psi(x_i)} \left[-log\, p_{Dec_\theta(z)}(x_i)\right] \\
&= \sum_i \mathbb{E}_{z \sim Enc_\psi(x_i)} ||Dec_\theta(z) - x_i||^2 + cst.
\end{aligned}
$$


μ, σ^2
f
(enc)
$\mu + \sigma Z$
g
(dec)
$Z : \mathcal{N}(0, \mathbf{I})$

Variational Auto-Encoder: Examples





Variational Auto-Encoder: The latent space

Example of the latent space in the VAE

Variational Auto-Encoder: Code Examples

```
class MLPVAE(nn.Module):
    def __init__(self, input_size, inter_size, latent_size):
        super().__init__()
        self.latent_size = latent_size
        self.encoder = nn.Sequential(...)
        self.m_sigma = nn.Linear(inter_size, latent_size)
        self.m_mu = nn.Linear(inter_size, latent_size)
        self.decoder = nn.Sequential(...)

    def decode(self, mu, sigma):
        z = torch.randn(mu.shape) * sigma + mu
        return torch.sigmoid(self.decoder(z))

    def generate(self):
        z = torch.randn(1, latent_size)
        return torch.sigmoid(self.decoder(z))

    def forward(self, x):
        r = self.encoder(x)
        mu, log_sigma = self.m_mu(r), self.m_sigma(r)
        y = self.decode(mu, torch.exp(0.5 * log_sigma))
        return y, mu, log_sigma
```

A short introduction to Generative Adverserial Network

Generative Adverserial Network

Objective

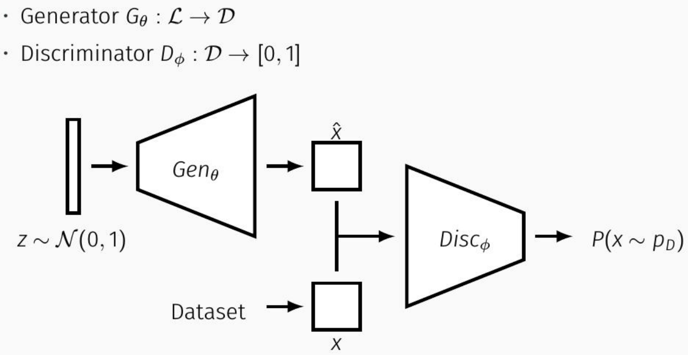
Find a generative model

- Classical approach: learn a distribution
- Idea: Sampling directly on latent space (z) and evaluate if the generated images is likely to be a correct image

Principle

- Find a good generative model where generated samples cannot be discriminated from real samples

- A dataset of true samples x (real)
- A generator G (a variational decoder), that sample from $p_g(z)$ and generate data (fake data)
- A discriminator D that disciminates real data from generated ones



Training GAN

- Alternate Optimisation
- Optimize D to discriminate fake from generated
- Optimize G to desceive D

Objective:

$$Min_G \, Max_D \mathbb{E}_{x \sim data} [\log D(x)] + \mathbb{E}_{z \sim p_g(z)} [\log(1 - D(G(z)))]$$

Generative Adverserial Network: Training GAN

Algorithm 1 Training GAN

for n iterations **do**

  for k steps **do** ▷ Discriminator update loop

    ◦ Sample $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ from prior noise $p_g(z)$

    ◦ Sample $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ from data distribution $p_{data}(x)$

    ◦ Update D by gradient descent:

$$\nabla_{\theta_D} - \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^i\right) + log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$$

  end for ▷ Generator update

  ◦ Sample $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ from prior noise $p_g(z)$

  ◦ Update D by gradient descent:

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$$

end for

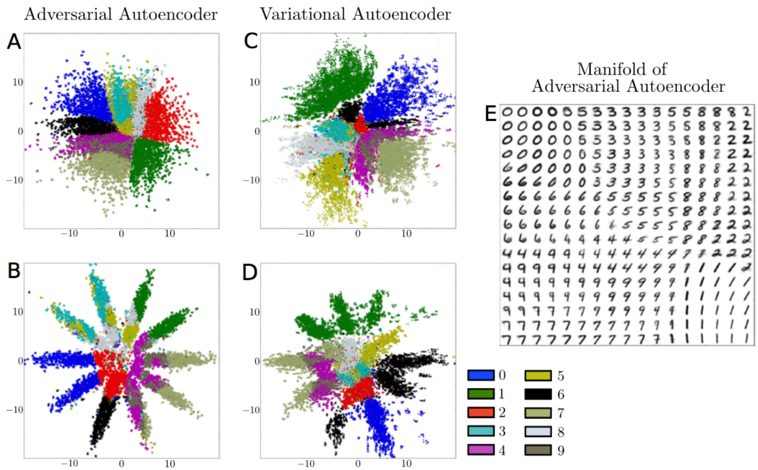


Figure 2: Comparison of adversarial and variational autoencoder on MNIST. The hidden code **z** of the *hold-out* images for an adversarial autoencoder fit to (a) a 2-D Gaussian and (b) a mixture of 10 2-D Gaussians. Each color represents the associated label. Same for variational autoencoder with (c) a 2-D gaussian and (d) a mixture of 10 2-D Gaussians. (e) Images generated by uniformly sampling the Gaussian percentiles along each hidden code dimension **z** in the 2-D Gaussian adversarial autoencoder.

Generative Adverserial Network: examples

Generative Adverserial Network: Training stability

Stability issues
If P_r (probability of real images) and P_g (on generated images) are not in the same manifolds :

- It exists a perfect discriminator
- End of optimization (but bad generator)

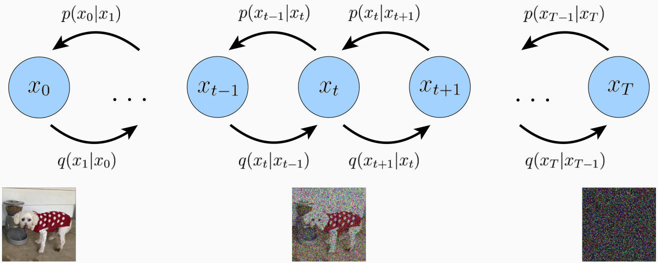
Stability issues
If P_r (probability of real images) and P_g (on generated images) are not in the same manifolds :

- It exists a perfect discriminator
- End of optimization (but bad generator)

Stabilizing training through regularization

- Improved training of Wasserstein GANs (Gulrajani, Ahmed, Arjovsky, Dumoulin, Courville 17)
  $\rightarrow$ using regularisation based on the wasserstein distance (with weights clipping)
- Stabilizing Training of Generative Adversarial Networks through Regularization (Roth, Lucchi, Nowozin, Hofmann, 17)

Diffusion models



Intuition beyond diffusion model

- Sampling gaussian noise
- Learn a denoising function $p(x_{t-1}|x_t)$
- Repeat the denoising step

What models are based on

- Stable diffusion
- Dall-E (since second version, first version was a GAN)
- Midjourney

[2]The image was taken from `https://calvinyluo.com/2022/08/26/diffusion-tutorial.html`