Fondamentaux de l'Apprentissage Automatique
Variantional Auto-Encoders & Denoising Diffusion Models

Lecturer: Yann Chevaleyre
Scribe: Linghao Zeng

Lecture n°11 #
08/12/2023

1 Variational Auto-Encoders

1.1 Auto-encoders

Standard auto-encoders learn to reduce the dimensionality of examples in a dataset. As shown in Fig. 1, the network is divided into two components : the encoder and the decoder. The encoder is responsible for compressing the input image into a lower-dimensional latent space, and the decoder works to reconstruct the input data from the latent space. The output is the reconstruction of the original input image, which the autoencoder tries to make as accurate as possible.

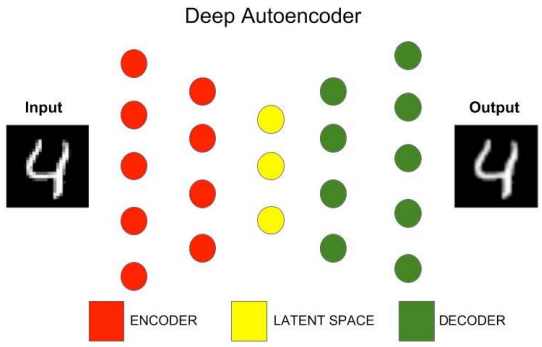

FIGURE 1 – Structure of standard auto-encoders.

Formally, we denote the encoder as $\text{Enc}_\varphi(x) = z$ parameterized by φ (a.k.a., weights), which maps the original input data x into a latent representation z. The decoder parameterized by ϕ, denotes as $\text{Dec}_\phi(z) = \tilde{x}$, maps z to the reconstructed data $\tilde{x}$.

Take one data point for $x_i \in \mathbb{R}^d$ for example, the encoder maps x_i into $z_i \in \mathbb{R}^k$, $k << d$, and the decoder reconstruct $\tilde{x}_i \in \mathbb{R}^d$ from z_i. To train this :

$$\underset{\varphi,\phi}{\text{argmin}} \sum_{i=0}^{N} \|x_i - \tilde{x}_i\|^2$$

where $z_i = \text{Enc}_\varphi(x_i)$ and $\tilde{x}_i = \text{Dec}_\phi(z_i)$.

Let's look at this process more intuitively, as shown in Fig. 2. the red dots represent original high-dimensional data points, and the green line is the lower-dimensional latent space learned by the encoder. The blue dots are reconstructions from the decoder, mapped back onto the green line, showing how the encoder-decoder network compresses and then reconstructs the data with some loss of detail.

Denoising auto-encoders is a kind of variant of standard auto-encoders.It aims to reconstruct a clean input from a noisy one. The objective is to minimize the reconstruction error, taking into account the added noise ϵ_i. Train as :

$$\underset{\varphi,\phi}{\operatorname{argmin}} \sum_i \|x_i - \mathrm{Dec}_\phi(\mathrm{Enc}_\varphi(x_i + \epsilon_i))\|^2$$

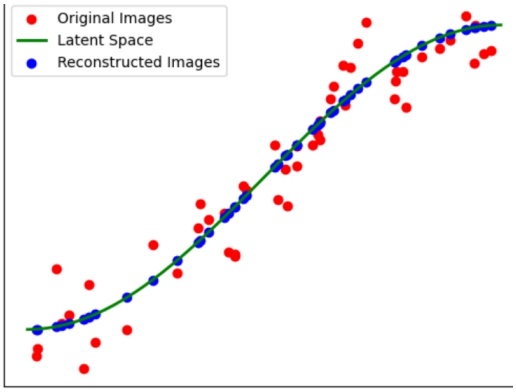

FIGURE 2 – A visualized example of auto-encoders.

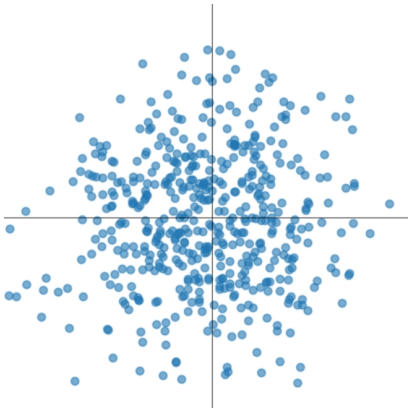

FIGURE 3 – A Gaussian distribution representing the latent space in an auto-encoder.

1.2 Variantional auto-encoders

Variational auto-encoders (VAE) learn to reduce the dimensionality of examples in a dataset and produce for each example x_i a distribution of latent vectors z_i such that overall, latent vectors $z_1 \dots z_N$ are normally distributed.

As illustrated in Fig. 4,

— **Encoding (Inference)** : an input image is transformed into a lower-dimensional latent space representation. The encoder produces two parameters : mean μ and standard deviation σ, which define a Gaussian distribution. This is achieved through the approximate posterior probability $q_\phi(z \mid x)$, representing the distribution of potential latent vectors z that could have generated the input x.

— **Decoding (Generative)** : By sampling from the late nt distribution, latent vectors z are fed into the decoder to reconstruct the input image $\tilde{x}$. The reconstruction is done by the decoder $p_\theta(x \mid z)$, which represents the likelihood of obtaining the original input x from the latent vector z.

We have

$$\begin{aligned} p_\theta(x \mid z) &= \mathcal{N}(x \mid \mu_z, I_d) \\ &= \mathcal{N}(x \mid \mathrm{Dec}_\theta(z), I_d) \text{ (because } \mu_z = \mathrm{Dec}_\theta(z)) \end{aligned}$$

and $x = \mathrm{Dec}_\theta(z) + \epsilon, \epsilon \sim \mathcal{N}(0, I)$.

1.2.1 Training the decoder alone

Assume we have a dataset $\{(x_i, z_i)\}_{i=1\dots N}$, where $x_i \in \mathbb{R}^d, z_i \in \mathbb{R}^k, k < d$.

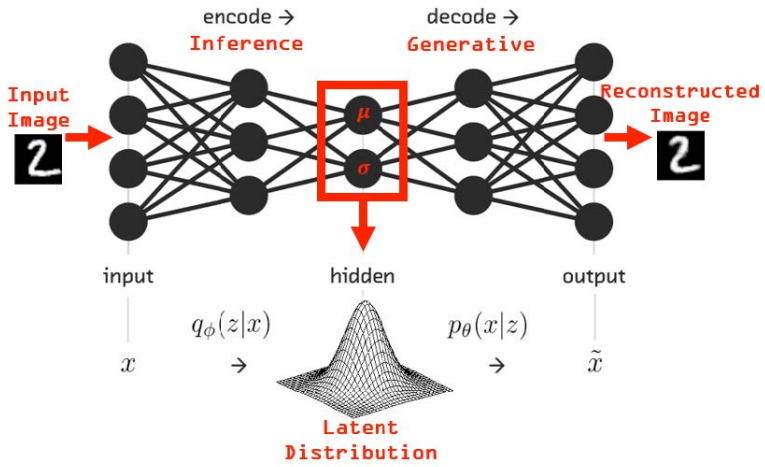

FIGURE 4 – Structure of variational auto-encoders.

$$\begin{aligned}\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log P_\theta\left(x_i \mid z_i\right) &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \mathcal{N}\left(x_i \mid \operatorname{Dec}_\theta\left(z_i\right), I\right) \\ &= \underset{\theta}{\operatorname{argmax}} \sum -\frac{1}{2}\left\|x_i - \operatorname{Dec}_\theta\left(z_i\right)\right\|^2 \\ &= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|x_i - \operatorname{Dec}_\theta\left(z_i\right)\right\|^2\end{aligned}$$

Of course, in practice we only have access to $\{x_i\}_{i=1\ldots N}$. Thus, we will train the encoder to learn the distribution of z for each x_i.

1.2.2 Understanding $p(x)$ and deriving the ELBO

We have $z_i \sim \mathcal{N}(0, I_k)$, I_k is $\mathbb{R}^{k\times k}$ identify matrix.

$$p_\theta(x) = \int_{\mathbb{R}^k} p_\theta(x \mid z) p(z) dz \xrightarrow[\text{is to train}]{\text{the goal}} \hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta\left(x_i\right)$$

Unfortunately, this integral is hard to estimate because for almost all $z \sim p(z)$, we have $p_\theta(x \mid z)$ close to zero.

Idea : apply the ELBO which requires $q_x(z) = q(z \mid x)$.
To represent $q(z \mid x)$, we will use an encoder,

$$x \xrightarrow{\text{Enc } \phi} \begin{cases}\mu_\phi(x) \\ \sigma_\phi(x), \text{ a } k \times k \text{diagonal matrix}\end{cases}$$

$q_\phi(z \mid x) = \mathcal{N}(z \mid (\mu_\phi(x), \sigma^2_\phi(x))$, where $\mu_\phi(x) \in \mathbb{R}^k$, $\sigma_\phi(x) \in \mathbb{R}^{k\times k}$

ELBO objective :

$$\begin{aligned}
\log p_\theta(x) &\geqslant \mathrm{ELBO}\,(x, \theta, q_\phi) \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right] \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln p_\theta(x \mid z) - \ln \frac{q_\phi(z \mid x))}{p(z)}\right] \\
&= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln p_\theta(x \mid z)\right] - KL\left(q_\phi(z \mid x), p(z)\right) \\
&= \mathbb{E}_{z\sim q_\phi(|x)}\left[-\frac{1}{2}\left\|x - \mathrm{Dec}_\theta(z)\right\|^2\right] - \underbrace{KL\left(q_\phi(z \mid x), q(z)\right)}_{\frac{1}{2}\left\|\mu_\phi(x)\right\|^2 + \mathrm{tr}\left(\sum_\phi(x)\right) - k - \log|\sum_\phi(x)|} + \mathrm{cst}
\end{aligned}$$

Reparametrization Trick :

— **Problem** : ELBO will not backpropagate on ϕ (encoder) because of $z \sim q_\phi$
$\Rightarrow$ can't compute $\nabla_\phi \mathrm{ELBO} \Rightarrow$ no training possible

— **Trick** : $z \sim q_\phi(\cdot \mid x) = \mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x)$ is distributed identically to $z' = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon$, where $\varepsilon \sim \mathcal{N}(0, 1)$

The computational graph is illustrated in Fig. 5.

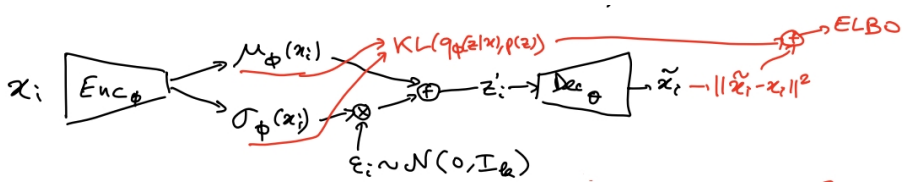

FIGURE 5 – Computational graph with the reparametization trick.

Then we can train this by variational EM :

1. Init $\hat{\phi} and \hat{\theta}$
2. $\hat{\phi} \leftarrow \underset{\phi}{\mathrm{argmax}}\, \mathrm{ELBO}(\hat{\phi}, \theta)$
3. $\hat{\theta} \leftarrow \underset{\theta}{\mathrm{argmax}}\, \mathrm{ELBO}(\phi, \hat{\theta})$

The procedure alternates repeatedly between performing step 2 and step 3 until convergence.

2 Denoising Diffusion Models

Diffusion models learn to create data by reversing a diffusion process, which gradually adds noise to the data until it turns into a Gaussian distribution. These models are trained to denoise data, iteratively learning to reconstruct the original data from the noisy distribution through a series of learned reverse diffusion steps. The denoising process is illustrated in Fig. 6. The followings are the algorithms for the training and sampling, respectively.

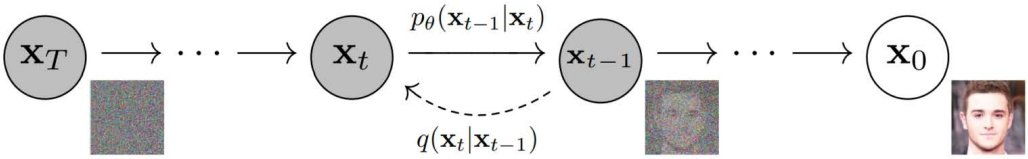

FIGURE 6 – Denoising of diffusion model.

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on : $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t \right) \right\|^2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T$ **to** 1 **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta (\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

We have :
$q(x_0)$: distribution of images in the dataset
$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t \mid \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t I_d\right)$
$x_t = x_{t-1}\sqrt{1-\beta_t} + \beta_t \cdot \epsilon_t$
$\epsilon_t \sim \mathcal{N}(0, I), \beta_t \in]0, 1[, \beta_1 < \beta_2 < \beta_3 \dots$
$q(x_1 \dots x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$ (chain rule of probability)

Observation :
— for simplicity, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

$$x_t = \sqrt{\alpha_t} \cdot x_{t-1} + \sqrt{1-\alpha_t} \cdot \varepsilon_{t-1}$$
$$x_{t-1} = \sqrt{\alpha_{t-1}} \cdot x_{t-2} + \sqrt{1-\alpha_{t-2}} \cdot \varepsilon_{t-2}$$
$$\vdots$$
$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\alpha_t} \cdot \varepsilon$$
$$q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\bar{\alpha}_t} x_0; (1-\bar{\alpha}_t) I_d\right)$$

— $p_\theta(x_0 \dots x_T) = p(x_T) \cdot \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$
Assume $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \sum_\theta(x_t, t)\right)$

Ideally, to learn $p_\theta(x_{t-1} \mid x_t)$, we would like to compute $q(x_{t-1}, x_t)$, which is hard.
Recall conditional Bayes rule $P(A \mid BC) = \frac{P(B|AC)P(A|C)}{P(B|C)}$
Instead, we compute

$$q(x_{t-1} \mid x_t, x_0) = \frac{q(x_t \mid x_t - 1, x_0) q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}$$
$$= \mathcal{N}(x_{t-1} \mid \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

where $q(x_t \mid x_t - 1, x_0)$, $q(x_{t-1} \mid x_0)$ and $q(x_t \mid x_0)$ are all known noramlly distribution.
$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t} \cdot (1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \cdot x_t + \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \cdot x_0$
$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t)$

$\Rightarrow \tilde{\mu}_t = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t)$

At the end, we count $q(x_{t-1}, x_t)$ to be close to $p_\theta(x_{t-1}, x_t)$
We want to maximize the ELBO, or minimize $-$ ELBO

$$
\begin{aligned}
-\text{ELBO} &= \mathbb{E}_q\left[-\log \frac{p_\theta(x_0, \ldots, x_T)}{q(\underbrace{x_1, \ldots, x_T}_{\text{latent variables}} \mid x_0)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(x_T) - \sum_{t\geqslant 1} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(x_T) - \sum_{t\geqslant 2} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1}, \underbrace{x_0}_{\text{does not change the value of } q(\cdot|\cdot)})} - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)}\right] \\
&\quad \text{(Conditional Bayes rule)} \\
&= \mathbb{E}_q\left[-\log p_\theta(x_T) - \sum_{t\geqslant 2} \log \left(\frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}\right) - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)}\right] \\
&= \mathbb{E}_q\left[-\log \frac{p_\theta(x_T)}{q(x_T \mid x_0)} - \sum_{t\geqslant 2} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} - \log p_\theta(x_0 \mid x_1)\right] \\
&= \mathbb{E}_q\left[\underbrace{KL(q(x_T \mid x_0), p_\theta(x_T))}_{L_T = cste} + \sum_{t\geqslant 2} \underbrace{KL(q(x_{t-1} \mid x_t, x_0), p_\theta(x_{t-1} \mid x_t))}_{L_{t-1}} - \underbrace{\log p_\theta(x_0 \mid x_1)}_{L_0}\right]
\end{aligned}
$$

Let us focus on L_{t-1} :

$$
\begin{aligned}
L_{t-1} &= KL\left(\underbrace{q(x_{t-1} \mid x_t, x_0))}_{\mathcal{N}(x_{t-1}|\tilde{\mu}_t, \tilde{\beta}_t I)}, \underbrace{p_\theta(x_{t-1} \mid x_t)}_{\mathcal{N}(x_{t-1}|\mu_\theta(x_t), \tilde{\beta}_t I)}\right) \\
&= \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t)\|^2 \times cste
\end{aligned}
$$

Recall that : $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \cdot \epsilon_t\right)$
Let us define :$\mu_\theta(x_t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \cdot \epsilon_\theta(x_t, t)\right)$, then

$$
\begin{aligned}
L_{t-1} &= \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \times cste \\
&= \left\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon_t, t\right)\right\|^2 \times cste
\end{aligned}
$$