Autoencoders and Variationnal Autoencoders : a brief introduction

LITO group meeting

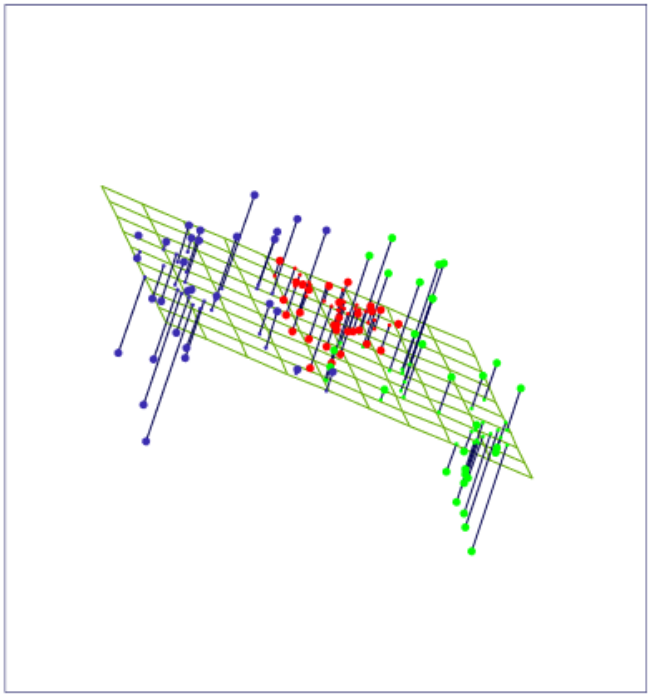
Nicolas Captier

May 19, 2021

Plan

PCA : minimum-error formulation

$X = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times p}$ (N centred observations)



We look for the best representation of our dataset in a hyperplane of dimension q :

$$\begin{cases} z = V_q^T x & \text{(encoding)} \\ x' = V_q z & \text{(decoding)} \end{cases}$$

*with $V_q \in \mathbb{R}^{p \times q}$ an orthogonal matrix

Minimize the reconstruction loss : $\min_{V_q} \sum_{i=1}^{N} ||x_i - V_q V_q^T x_i||^2$

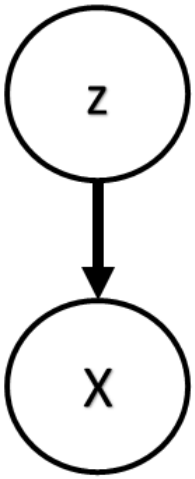
Solution : $V_q = U_q \Lambda_q^{\frac{1}{2}}$ (U_q , Λ_q first q eigenvectors and eigenvalues of the empirical variance)

PPCA : a gaussian probabilistic model

We now consider x and z being **random variables**. The observations $(x_1, ..., x_N)$ samples from the following **probabilistic model** :

$$x = V_q z + \epsilon \quad , \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I_p)$$

$$\begin{cases} p(z) \sim \mathcal{N}(0, I_q) & \text{(prior distribution)} \\ \\ p(x|z) \sim \mathcal{N}(V_q z, \sigma^2 I_p) & \text{(marginal likelihood)} \end{cases}$$



Integrating over z, we obtain the **observed likelihood** :

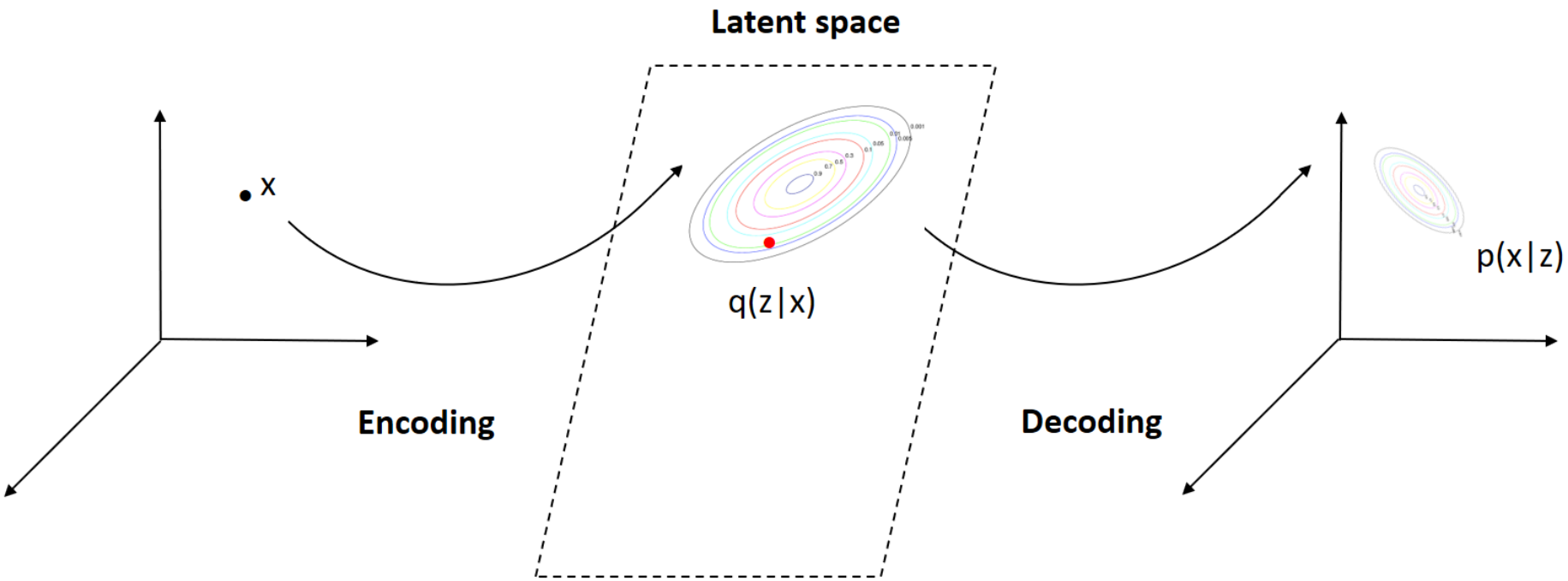
$$p(x) \sim \mathcal{N}(0, V_q V_q^T + \sigma^2 I_p)$$

MLE estimators : $\hat{V}_q^{MLE}, \hat{\sigma}^{MLE} \in \underset{V_q, \sigma}{argmax}\, \mathcal{L}(x_1, ..., x_N)$

Solution : $\hat{V}_q^{MLE} = U_q (\Lambda_q - \sigma^2 I_q)^{\frac{1}{2}}$

A change of paradigm : from points to distributions

Observations x_i are no longer associated with single points z_i but with probability distributions $p(z|x_i)$



*In this simple case we can explicitly derive the posterior distribution $p(z|x) \propto p(x|z)p(z)$. For more complicated probabilistic model it will not be the case. **We will need a variational approach !**

Plan

A naive non-linear dimensionality reduction technique

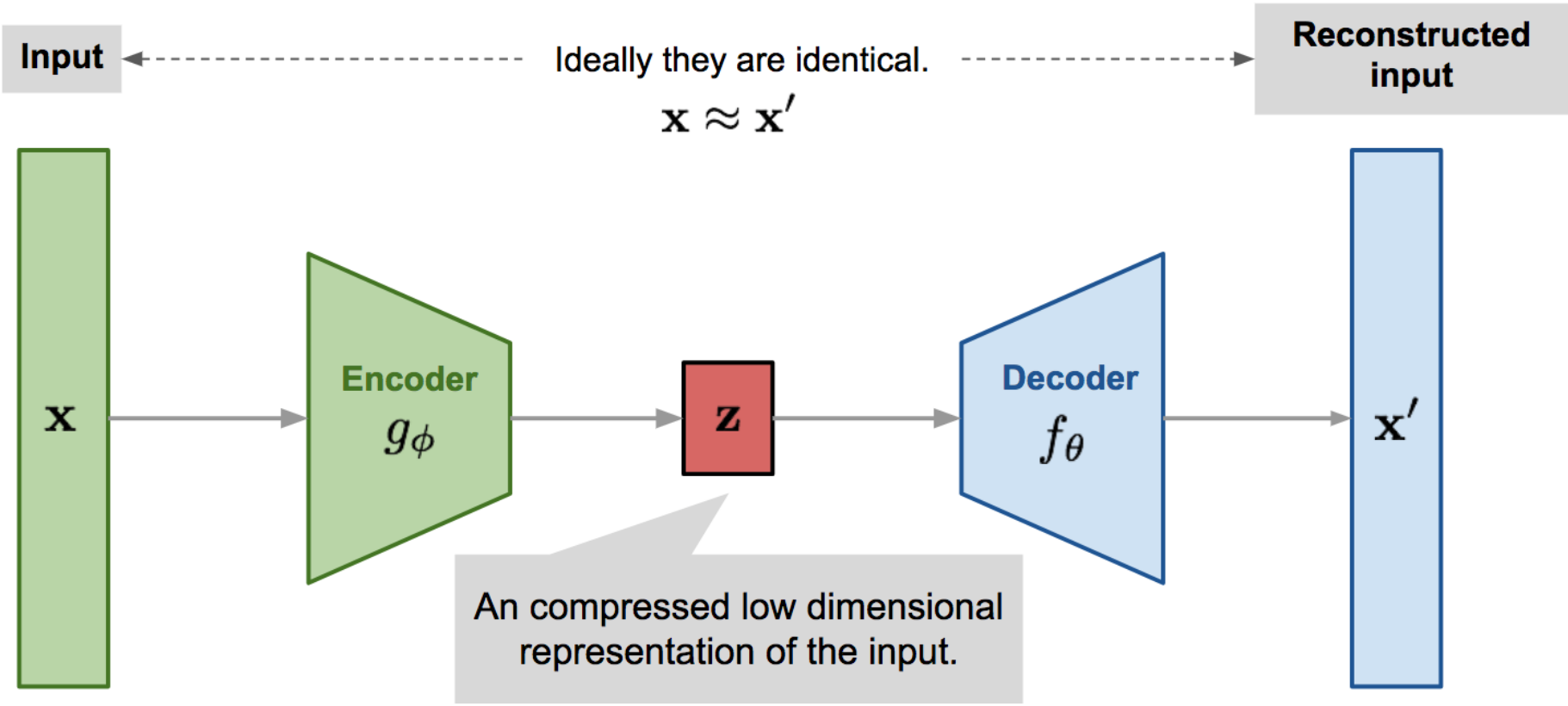
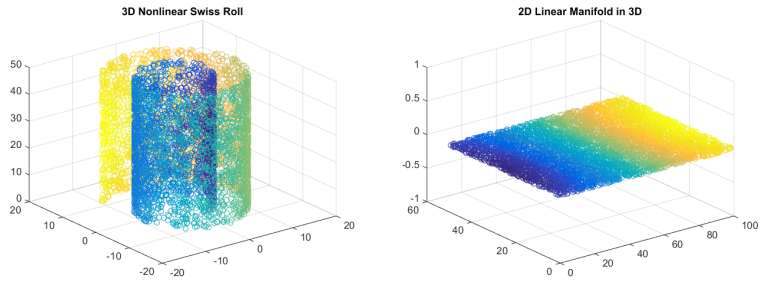


Figure 1: An undercomplete autoencoder

Reconstruction error : $\theta^*, \Phi^* \in \underset{\theta,\phi}{argmin} \frac{1}{n} \sum_{i=1}^{n} ||x_i - \overbrace{g_\Phi\left(f_\theta(x_i)\right)}^{x'}||^2$

Meaningful representation or excellent memorizing ?

We need to make sure that the autoencoder will learn **a meaningful representation**. We want to capture the **latent manifold structure** (i.e generalization of the hyperplane in the linear case).

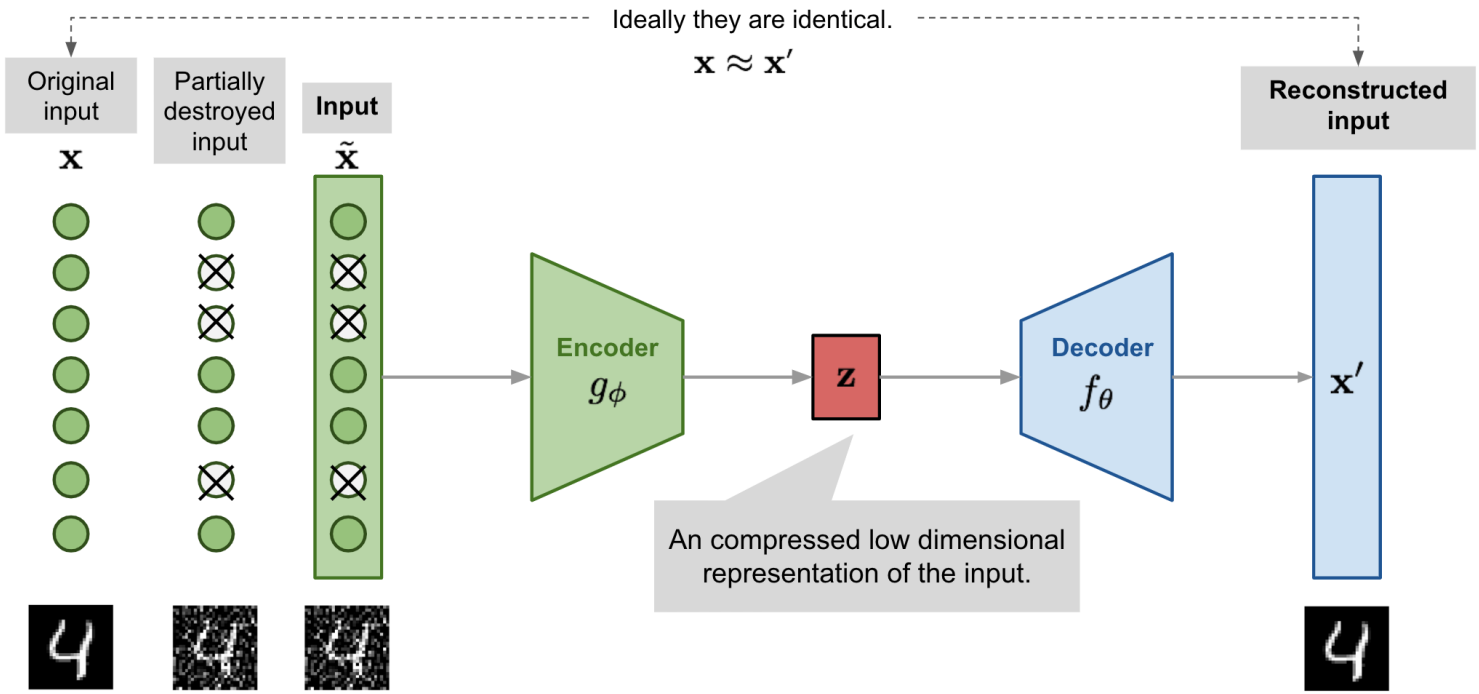


The reconstruction loss is not sufficient since it does not constrain the latent representation. We need to modify it !

Denoising autoencoders

A good representation should capture information robust to partial destruction of the input (e.g humans are able to recognize partially destroyed high-dimensional data such as images).

$$\theta^*, \Phi^* \in \operatorname{argmin} \frac{1}{n} \sum_{i=1}^{N} L(x_i, g_\Phi(f_\theta(\tilde{x}_i)))$$



*Corruption can take various forms : additive gaussian isotropic noise, drop-out noise, salt-and-pepper noise...

Contractive autoencoders

The robustness to small perturbations is ensured by an additional penalty :

Reconstruction loss + regularization

$$\theta^*, \Phi^* \in \underset{}{argmin} \frac{1}{n} \sum_{i=1}^{N} L(x_i, g_\Phi(f_\theta(x_i)) \quad + \quad \underbrace{\lambda ||\mathcal{J}_f(x)||_F^2}_{\text{penalizes the gradient w.r.t input}}$$

- We penalize cases where a small change in the input leads to a large change in the encoding space.
- we're essentially forcing the model to learn how to contract a neighborhood of inputs into a smaller neighborhood of outputs.

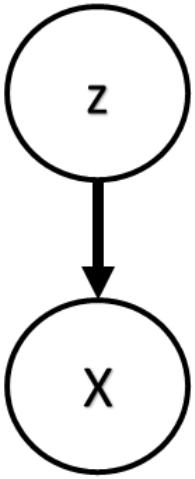
* Frobenius norm : $||A||_F^2 = \sum_{i,j=1}^{n} |a_{ij}|^2$

Plan

Here we go again : non-linear probabilistic model

Like with PPCA, $(x_1, ..., x_N)$ results from **a probabilistic generative model** based on a latent variable z.

$$\begin{cases} p(z) \sim \mathcal{N}(0, I_q) \quad \text{(prior distribution)} \\ p_\theta(x|z) \sim \mathcal{N}(f_\theta(z), \tilde{f}_\theta(z) I_p) \quad \text{(marginal likelihood)} \end{cases}$$



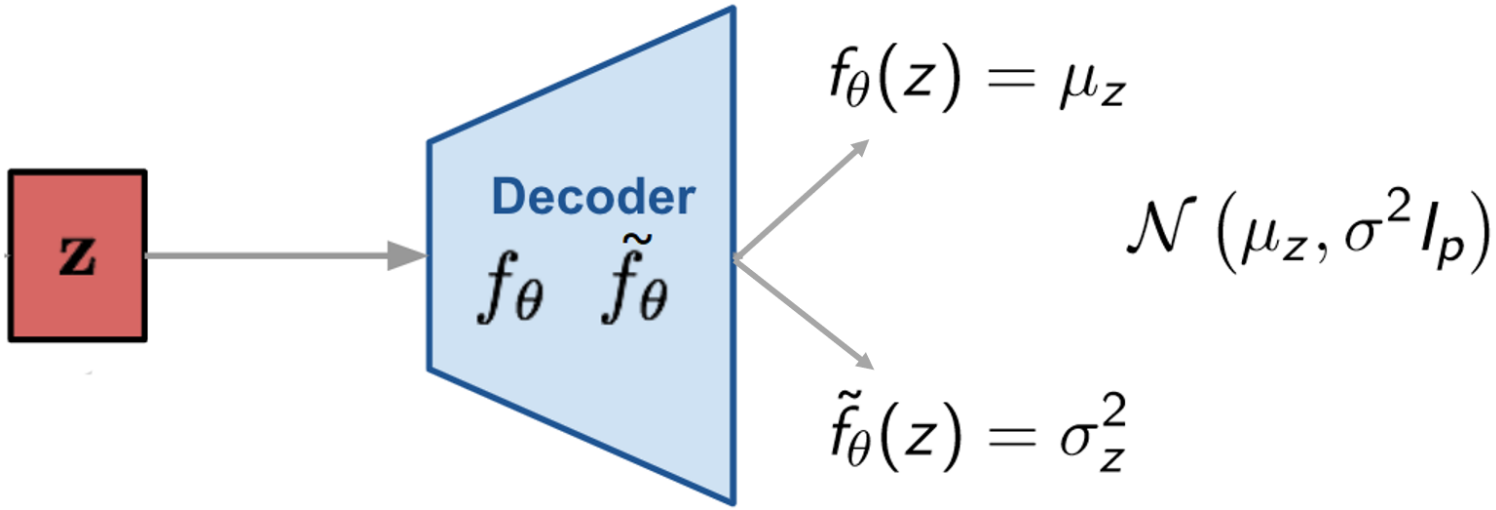


Figure 2: probabilistic decoder (neural network)

Encoder as a variational approximation

Main issue : the true posterior $p_\theta(z|x) \propto p_\theta(x|z)p(z)$ is intractable. The simple PPCA approach (i.e find the best θ^* with MLE and use $p_{\theta^*}(x|z)$ for encoding) is not a valid approach anymore.

Variational inference principle

- Approximate by a simple parametric distribution $q_\Phi(z|x)$.
- Optimize Φ to minimize the Kullback-Leibler divergence between $q_\Phi(z|x)$ and $p_\theta(z|x)$.

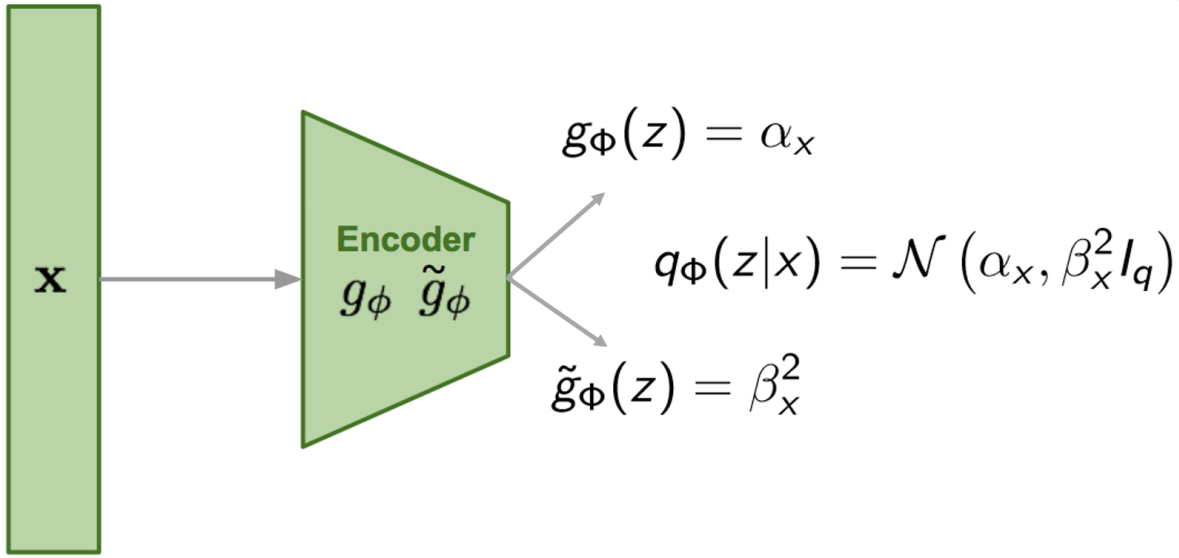
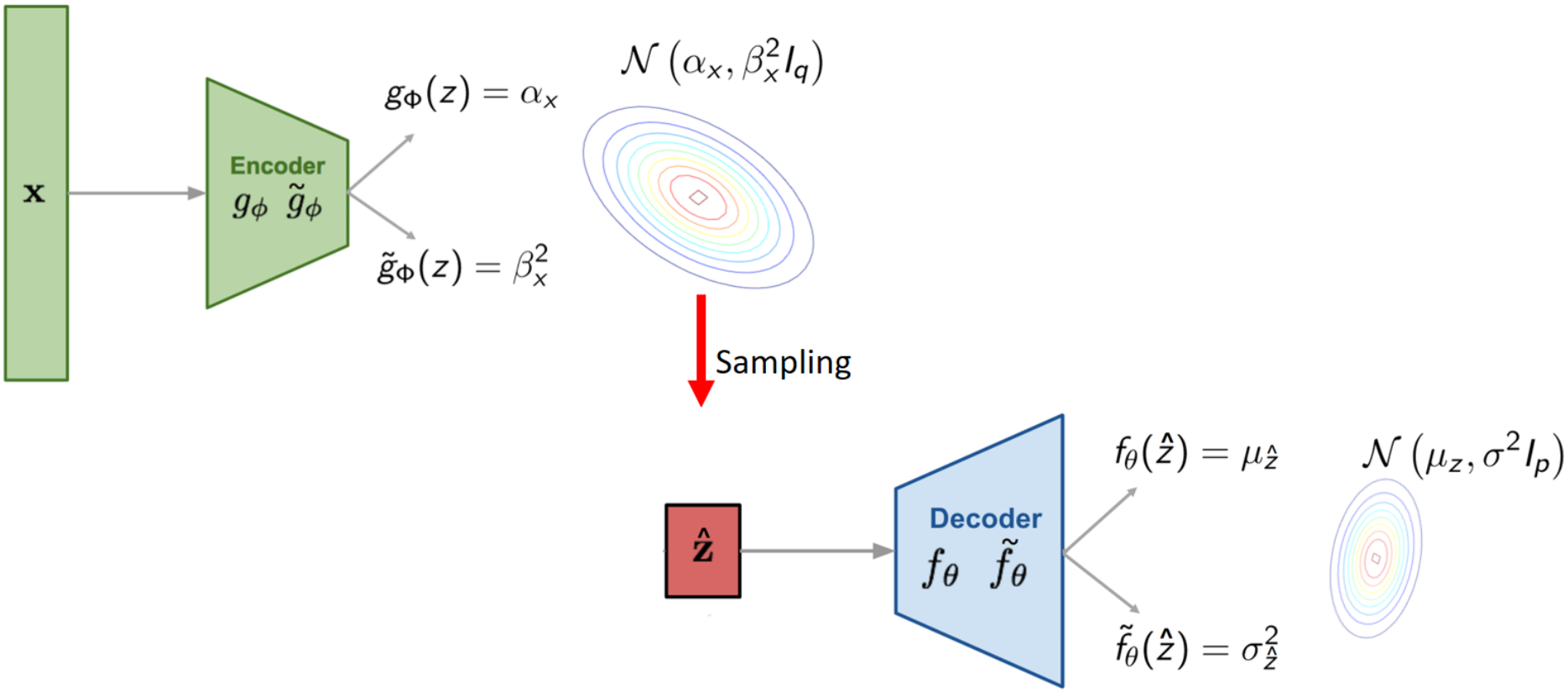


Figure 3: probabilistic encoder (variational approximation)

The big picture : encoding + decoding



- The output of the variational autoencoder is a probability distribution $p_\theta(x|\hat{z}_i)$ (with $\hat{z} \sim q_\Phi(z|x_i)$).
- Intuitively, we should expect the input/observation x_i to be associated with a strong probability value $\log p(x_i|\hat{z}_i)$.

Training objective for the variational autoencoder -I

Two interlinked objectives :

minimize $KL\left(q_\Phi(z|x)||p_\theta(z|x)\right)$ (find the best encoding)

maximize $\log p_\theta(x)$ (best fit the observations)

Evidence Lower Bound (ELBO)

$$KL\left(q_\Phi(z|x)||p_\theta(z|x)\right) = -\underbrace{\mathcal{L}(x,\theta,\Phi)}_{\text{ELBO}} + \log p_\theta(x)$$

- The KL divergence being positive, the ELBO is a lower bound for the log-likelihood
- Minimizing $\mathcal{L}$ w.r.t Φ we do minimize the KL divergence and look for the best variational approximation (for a fixed θ).

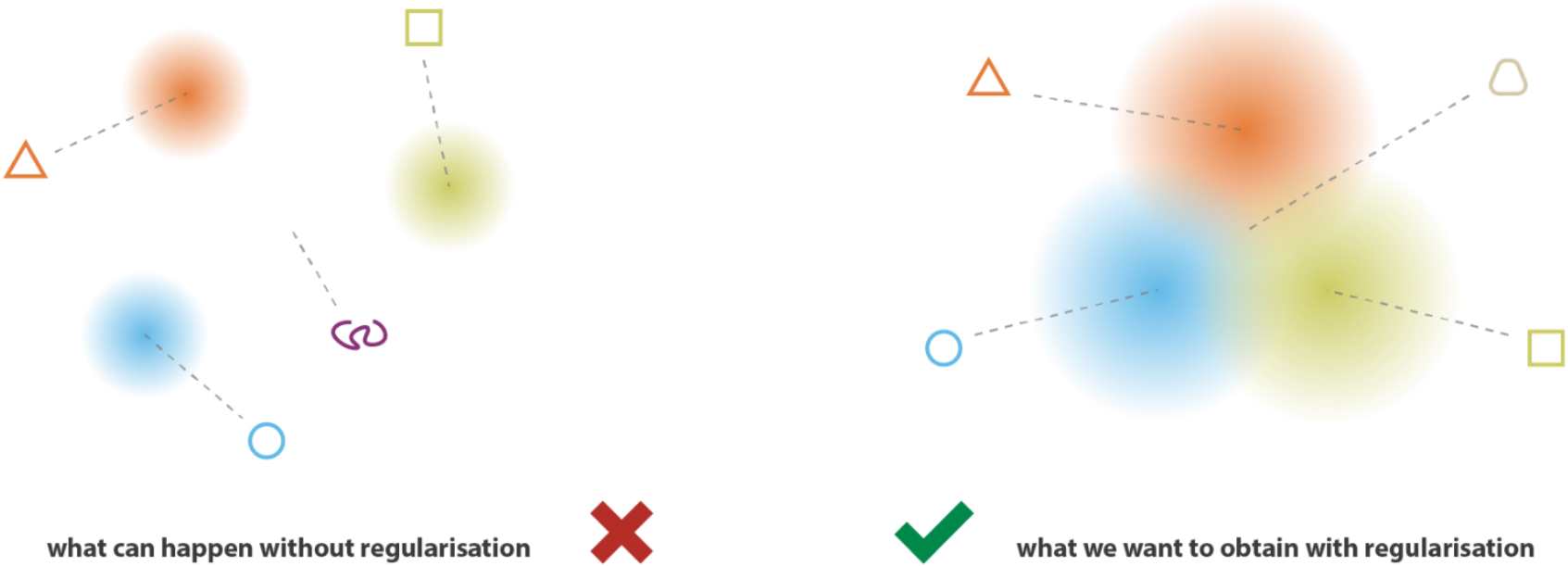
Final objective

$$\theta^*, \Phi^* \in \mathit{argmax}\, \mathcal{L}(x,\theta,\Phi)$$

Training objective for the variational autoencoder -II

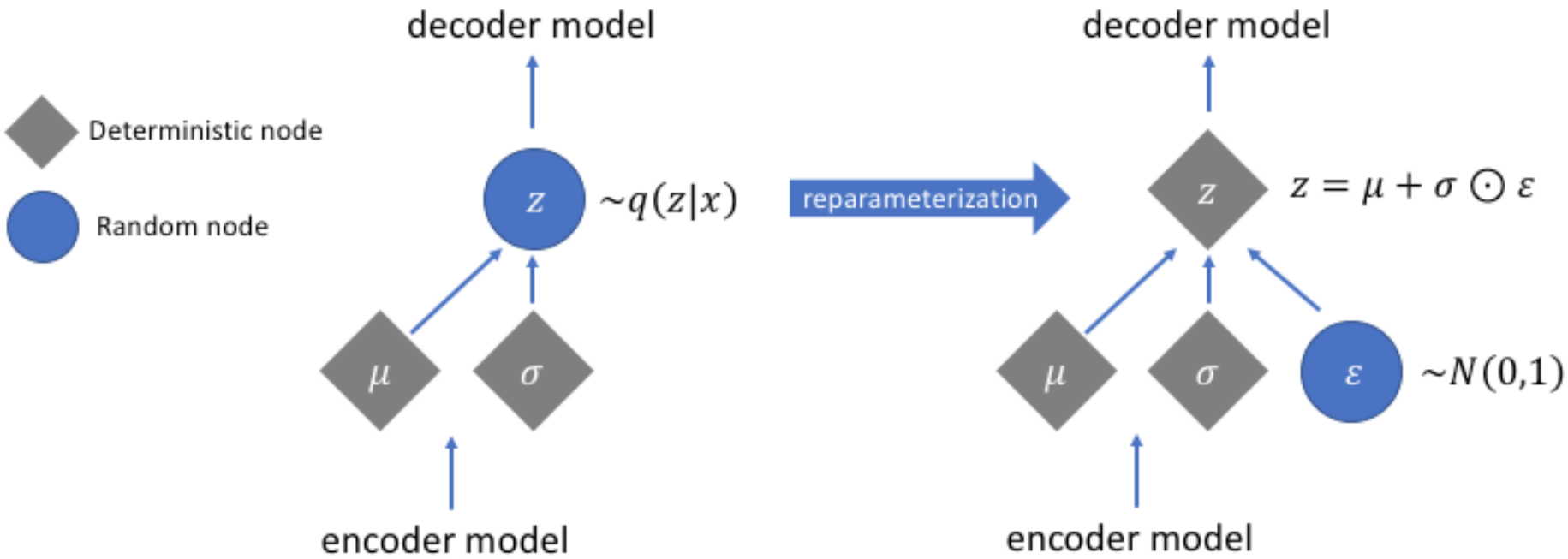
$$\max \mathcal{L}(x, \theta, \Phi) = \underbrace{-KL\left(q_\Phi(z|x)||p(z)\right)}_{\text{regularization term}} + \underbrace{\mathbb{E}_{q_\Phi(z|x)}[\log p_\theta(x|z)]}_{\text{reconstruction term}}$$

* In our gaussian case, the KL term can be computed analytically
* The expected term can be estimated by sampling several time $\hat{z} \sim q_\Phi(z|x)$ and computing the empirical mean.



The reparametrization trick

- In order to optimize the training objective, we would like to apply gradient descent optimization through the whole neural network system.
- The sampling step does not support backpropagation ! We need to decouple the stochastic part and the deterministic parts that contain the parameters we want to optimize.



A few personnal thoughts about VAEs

- In theory, the variational autoencoders are not bound to gaussian distributions. Any distribution which can be parametrized by a neural network and which is associated with a reparametrization trick should work.
- The isotropic gaussian prior $p(z) \sim \mathcal{N}(0, I)$ has a strong influence on the latent space. It could be sometimes far to naive. Sometimes, it is worth enforcing this regularization with β penalty.
- Contrary to GANs or simple autoencoders, VAEs come from a solid mathematical building that may allow for further developments and complexifications.

Though variational autoencoders have shown very promising results within the past 10 years, many obstacles remain to be overcome.

Non-exhaustive bibliography

- "Introduction to autoencoders" - Jordan (online post)
- "Understanding Variational Autoencoders (VAEs)" - Rocca (online post)
- "Auto-Encoding Variational Bayes" - Kingma , Welling 2014
- "What Regularized Auto-Encoders Learn from the Data-Generating Distribution" - Alain , Bengio 2014
- "Probabilistic principal component analysis" - Tipping , Bishop 1999
- "Extracting and Composing Robust Features with Denoising Autoencoders" - Vincent et al. 2008
- "Disentangling Disentanglement in Variational Autoencoders" - Mathieu et al. 2018