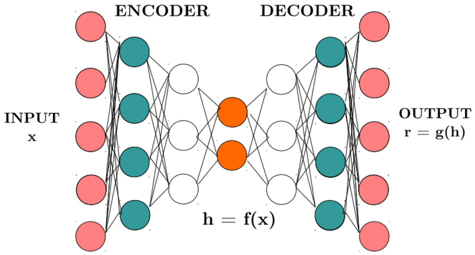
Variational Autoencoder

Matteo Boschini, Lorenzo Bonicelli

December 10, 2021

University of Modena and Reggio Emilia

An **autoencoder** is a feed-forward neural network that is trained to copy its input to its output.
The network may be viewed as consisting of two parts: an **encoder** $h = f(x)$ that produces a **hidden representation** and a **decoder** that produces a **reconstruction** $r = g(h)$.



Undercomplete Autoencoder

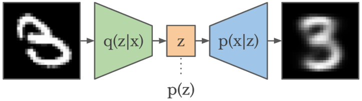
"Generative modeling" is an area of machine learning which deals with models of distributions $P(X)$, defined over datapoints $x \in X$, where X is usually a high-dimensional space.

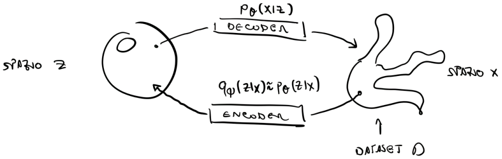
- In the case of images, X values which look like real images should get high probability, whereas images that look like random noise should get low probability.
- The aim is to synthesize **NEW** examples (e.g. MNIST digits images) that look like those already in a dataset, but not exactly the same.

Denoising and contractive autoencoders potentially learn the structure of a probability distribution $P(X)$... *but how can you sample new examples from them?*

Variational Autoencoder

A **Variational Autoencoder (VAE)** requires its hidden representation to abide by a well-behaved distribution $q(z)$ (e.g. **unit Gaussian**). We regard z as an auxiliary (latent) variable.





The VAE will:

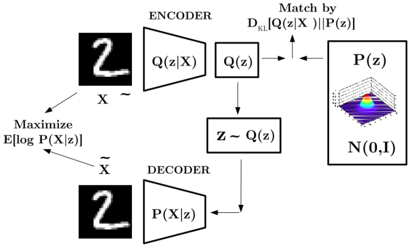
1. take an input data-point x and generate a hidden distribution $q(z|x)$;
2. Sample $z \sim q(z|x)$ and generate a probability distribution back in input space $p(x|z)$.

These distributions are parameterized by an **Encoder** ($q_\phi(z|x)$) and **Decoder** ($p_\theta(x|z)$) model respectively (with parameters ϕ and θ).

VAE Training Objective

The VAE optimizes the following objective:

$$\underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{Reconstruction Term}} - \lambda \underbrace{D_{KL}[q_\phi(z|x)||p(z)]}_{\text{Regularization Term}}$$

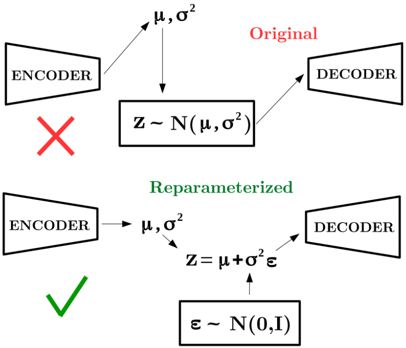


- The Reconstruction Term (e.g. BCE or MSE), is the typical autoencoder objective: what comes out of the decoder must mimic the initial input.
- The Regularization Term requires encoded examples to behave like a unit Gaussian ($\sim \mathcal{N}(0, I)$).

To model $q(z|x)$, the encoder produces 2 separate outputs, corresponding to μ and σ^2 of a Gaussian distribution.

However, we cannot backpropagate gradients through a sampling operation!

Instead, of computing $z \sim \mathcal{N}(\mu, \sigma^2)$, we **externalize the randomness** in a random variable $\varepsilon \sim \mathcal{N}(0, I)$ and compute z as $z = \mu + \varepsilon \cdot \sigma^2$.



One More Thing...

The variance (σ^2) output by the Encoder is strictly positive! The Encoder network must yield a positive value. We can do this by

1. taking the absolute value of the output;
2. applying the ReLU to the outpu;
3. **BEST!** assume that we are not predicting σ^2 but rather $\log \sigma^2$

The latter option maps $(-\infty, 0)$ into $(0, 1)$, thus leaving more room for the network to optimize and making the training process more stable.