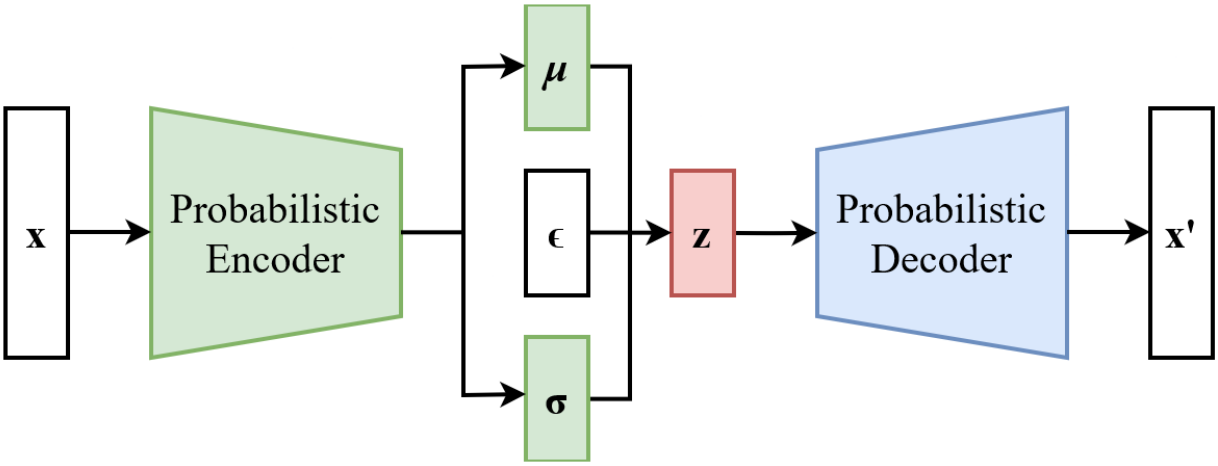
VAE + GAN Minichallenge

In this minichallenge, we pre-train a VAE that is then used to train a GAN.

VAE

The VAE (Variational Autoencoder) model encodes image data into latent space using convolutional layers. It uses 2D convolutional layers to progressively reduce the spatial dimensions of the input. Then, a decoder part is used with 2d deconvolutional layers to increase the spatial dimensions back to the original size. The interesting thing is that the model learns to generate images of the same style as the training data from random noise. The following picture shows the model architecture:



Mathematically, we want to maximize the likelihood of the data x(in the figure) given a parameterized probability distribution $p_\theta(x) = p(x|\theta)$. This distribution is usually chosen to be a Gaussian $N(x|\mu, \phi)$ which is parameterized by μ and ϕ respectively.

The probability of $p_\theta(x)$ can be found by integrating over the joint probability: $p_\theta(x) = \int_z p_\theta(x, z)dz$. This can be rewritten to $p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$ with the chain rule ($P(A \cap B) = P(B | A) * P(A)$). This can be modeled with the VAE model and its architecture. "z" is usually taken to be a finite-dimensional vector of real numbers. $p_\theta(x|z)$ is a Gaussian distribution. $p_\theta(x)$ is a mixture of Gaussian distributions.

- To optimize this model, one needs to know two terms: the "reconstruction error", and the Kullback–Leibler divergence. Both terms are derived from the free energy expression of the probabilistic model, and therefore differ depending on the noise distribution and the assumed prior of the data. For example, a standard VAE task such as IMAGENET is typically assumed to have a gaussianly distributed noise, however tasks such as binarized MNIST require a Bernoulli noise.

Data

Variational Autoencoders and Generative Adversarial Networks