TD-2 : CNN and VAE

Exercise 1 : Convolutional Neural Network

Question 1 :

Let consider a linear layer (with no bias). it takes as input $x \in \mathbb{R}^{10\times10\times3}$ an image and produce an intermediate representation $z^{(1)} \in \mathbb{R}^{10\times10\times20}$. How many trainable parameters (float) the model has (you can consider flattening input/ouput) ? What problem can we have if the input size of the image is high ?

Question 2 :

Let consider a CNN 2 dimenssional layer, it takes as input $x \in \mathbb{R}^{10\times10\times3}$ an image and produce 20 features maps, the kernel or filter size are 3×3. How many trainable parameters (float) the model has ?

Question 3 :

Considering the same architecture, what is the size of the features maps without considering padding ? If we consider a stride of 2 ?

Exercise 2 : VAE

Question 1 :

Show that $\log(p(x)) \geq \mathbf{E}_{z\sim q_\psi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left[q_\psi(z|x)\ ||p(z)\right]$

Question 2 :

What measures the KL divergence ? When it is equals to 0 ? Provide the associated formula for two distributions q and p

Question 3 :

Let consider $p(z)$ as the standard normal distribution (i.e $\mathcal{N}(0,1)$) and $q_\psi(z|x)$ a normal distribution parametrized by σ^2, μ, what is the expression of $D_{KL}\left(q_\psi(z|x)\ ||p(z)\right)$?

Question 4 :

What is the loss associated to the ELBO (gradient descent) considering $p(z)$ as the standard normal distribution (i.e $\mathcal{N}(0,1)$) and $q(z|x)$ a normal distribution parametrized by σ^2, μ.

Question 5 :

How can we approximate $\mathbb{E}_{z\sim q_\psi(z|x)}\left[-log\left(p_\theta(x|z)\right)\right]$

Question 6 :

Let assume (in addition) that we consider reparametrization trick evaluating only one sample (not a good monte-carlo estimation for). We consider q_{enc} an MLP producing a representation of x, q_μ, q_σ layers producing the mean and the variance for distribution $q_\psi(z|x)$, p_{dec} the decoder function (modeling $p_\theta(x|z)$). Rewrite the loss function :