Lab session:
Neural Priors for Images

Master ID3D – Julie Digne

October 16, 2025

In this lab we will see how to use neural priors to reconstruct images or denoise them

1 Libraries & software

Programming is done in Python, using the following libraries:

- Numpy
- Matplotlib (to visualize the evolution of the cost functions)
- Pytorch (for the neural part)
- PIL (for reading and saving images)
- A conda environment file \texttt{tp_ml.yml} is available on the course page.

2 Neural Prior for Reconstruction

We will first build a network capable of learning an image — a simple encoder-decoder with a few skip connections.

As input, we take a 16-channel noise image, and the network reconstructs a 3-channel image (thus, a slightly asymmetric architecture).

The proposed architecture is as follows:

- Input: 16 channels
- Down layer 1: 32 channels
- Down layer 2: 64 channels
- Down layer 3: 128 channels
- Down layer 4: 256 channels
- Up layer 4: 128 channels
- Up layer 3: 64 channels
- Up layer 2: 32 channels
- Up layer 1: 3 channels
- 2 skip connections at sizes 128 and 64. The skips are convolutions with kernel sizes 5 and 4, and the results are concatenated.

Additionally:

- Activations are leaky_relu (except the last one, which is a sigmoid)
- The "up" layers use transposed convolutions with kernel size 4, stride 2, and padding 1.

Implement the network and test simple image reconstruction.

3 Neural Prior for Inpainting

Randomly mask pixels in the input image and use the network to reconstruct the image.

Be careful: the loss should only take into account the known pixels.

4 Neural Prior for Denoising

Add Gaussian noise to the input image, and use the network to reconstruct the image.