ATIV - Neural Prior

Julie Digne

Master ID3D
LIRIS - CNRS
Équipe Origami

13/11/2024

Solving inverse problems

- Recover a signal from a set of measurements/observations (MRI...)

Problem statement

Find u a signal/image/function, knowing measures v

$$F(u) = v$$

With F a model of the measurement operator

Deep Image Prior [Ulyanov et al 2018]

Inverse Problems

We know only a degraded version of an image, we want to recover the original one.

Deep Image Prior [Ulyanov et al 2018]

Inverse Problems

We know only a degraded version of an image, we want to recover the original one.

ML + Inverse Problems

Learn the inverse transform from example

Deep Image Prior [Ulyanov et al 2018]

Inverse Problems

We know only a degraded version of an image, we want to recover the original one.

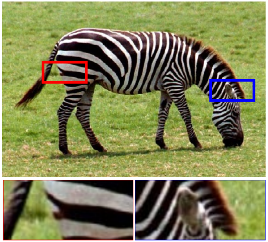
ML + Inverse Problems

Learn the inverse transform from example

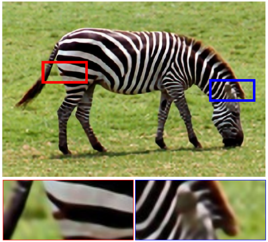
Deep Image Prior

Not all statistics need to be learned from databases, a lot is captured by the structure of generative convolutional nets.

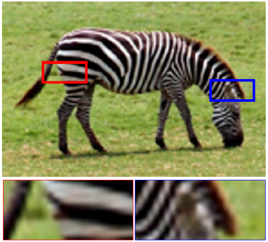
Inverse problem: Super-resolution



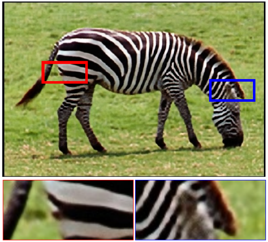
(a) Ground truth



(b) SRResNet [36], **Trained**



(c) Bicubic, **Not trained**



(d) Deep prior, **Not trained**

Deep generator formulation

Deep generator

A network parametered by θ which maps an input code vector z to an image x:

$$x = f_\theta(z)$$

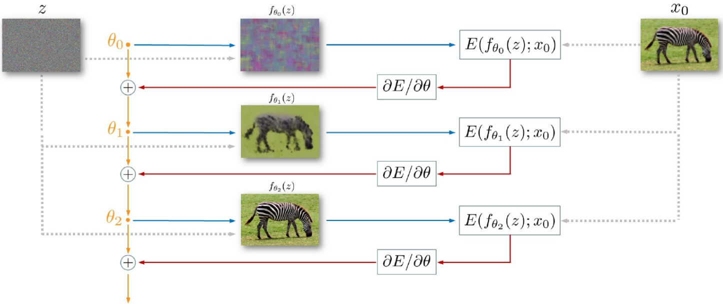
Usual DL approach

Knowledge about the distribution $p(x)$ of x is encoded in θ which should be optimized wrt to a database.

Deep prior approach [Ulyanov 2018]

A significant amount of information is encoded in the structure of the generator network, even without training its parameters.

Principle



[Ulyanov 2018]

Energy formulation

Inverse problem

x_0 is observed, one wants to find x close to x_0 but *better*:

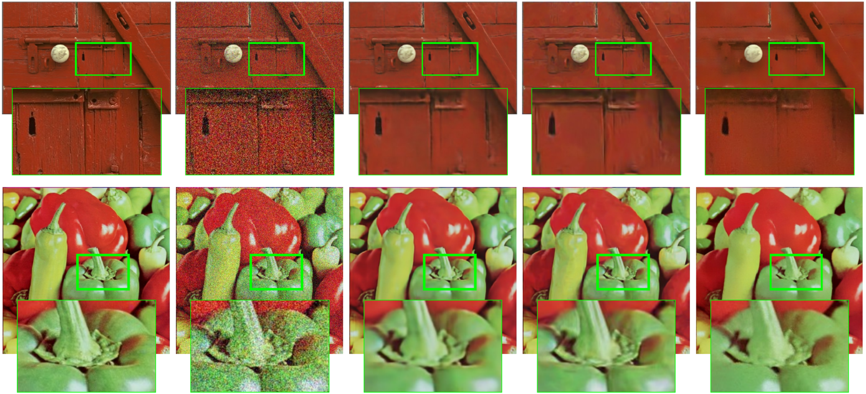
$$x^* = argmin_x E(x, x_0) + R(x)$$

- E is an energy (e.g. $E(x, x_0) = \|x - x_0\|^2$)
- R is a regularization term (e.g. $R(x) = \|x\|_2$, $R(x) = TV(x)$).

Deep Prior = regularizer

$$\theta^* = argmin_\theta E(x, f_\theta(z) \text{ with } x^* = f_{\theta^*}(z)$$

Denoising



(a) GT (b) Input (c) Ours (d) CBM3D (e) NLM

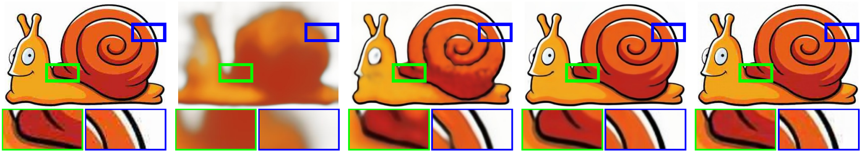
[Ulyanov 2018]

Inpainting

(a) Corrupted image (b) Global-Local GAN [28] (c) Ours, LR = 0.01 (d) Ours, LR = 10^{-4}

[Ulyanov 2018]

JPEG artefact removal



Corrupted	100 iterations	600 iterations	2400 iterations	50K iterations

[Ulyanov 2018]

- Stop before overfitting!

More complex inpainting

(a) Input (white=masked)

(b) Encoder-decoder, depth=6

(c) Encoder-decoder, depth=4

(d) Encoder-decoder, depth=2

(e) ResNet, depth=8

(f) U-net, depth=5

[Ulyanov 2018]

Lab work

- Train your own deep prior network
- Encoder-decoder architecture with two skip connections
- First application: plain image reconstruction
- Applications: denoising and inpainting

Convolutional Neural Networks

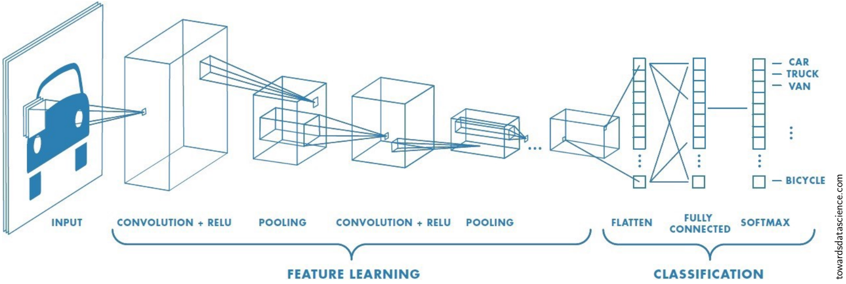
Shared parameters

Dropping fully connected layers, CNN use convolutions by kernels with weights independent of the image location. These weights are optimized during training.

Convolutional Neural Networks

Shared parameters

Dropping fully connected layers, CNN use convolutions by kernels with weights independent of the image location. These weights are optimized during training.



Convolution layer parameters

- Kernel size: controls the locality of the kernel
- Padding: increases the size of the input
- Dilatation: aggregates values from every n pixels where n is the dilatation. (eq to set some weights in the kernel to 0).
- Stride: performs the convolution centered every n pixels where n is the stride.

Visualization

`https://ezyang.github.io/convolution-visualizer/index.html`