How Machine Learning can help to automate processing tasks? An example with image denoising

François-Xavier Dupé

(Qarma – LIS – Aix-Marseille Université, France)

Machine Learning Seminary @ LAM

What is Machine Learning?

The aim of Machine Learning is to build a mathematical function which solve a human task.

What is Machine Learning?

The aim of Machine Learning is to build a mathematical function which solve a human task.

Today tasks include

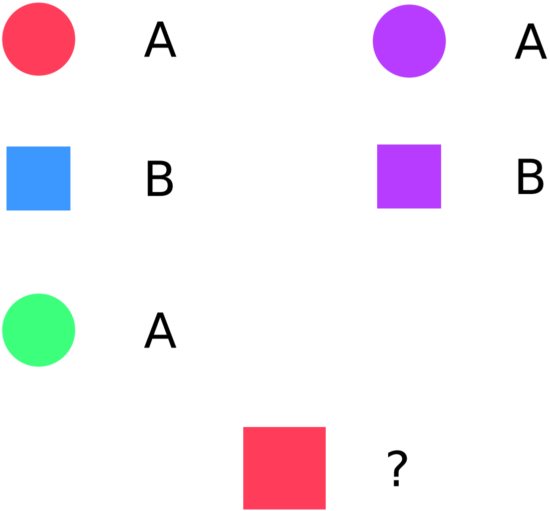
- regression/classification;
- representation (or feature) learning;
- ...

Today's talk

1. Global introduction to machine learning
2. Small introduction to deep learning
3. Learning to denoise image
4. Some extensions

Today's talk

Let's play a little



Some mathematical context

$(x_i, y_i)_{i\in\{0,\dots,n\}} \subseteq \mathcal{X} \times \mathcal{Y}$ an observation sample

- x_i are the input data.
- y_i are the target data.
- x_i and y_i may be of different nature, e.g. $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$.
- n can be very large…

Aim of machine learning

Main problem,

$$\text{find } f : \mathcal{X} \to \mathcal{Y} \text{ such that } \forall i \quad f(x_i) = y_i \ . \tag{P}$$

Remarks:

Aim of machine learning

Main problem,

$$\text{find } f : \mathcal{X} \to \mathcal{Y} \text{ such that } \forall i \quad f(x_i) = y_i \ . \tag{P}$$

Remarks:

- General setting impossible!

Aim of machine learning

Main problem,

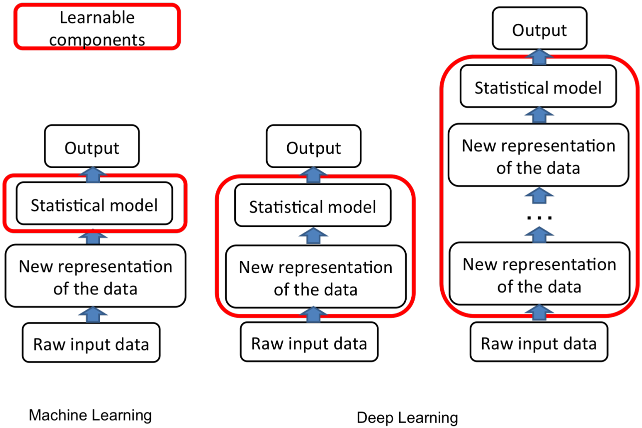
$$\text{find } f : \mathcal{X} \to \mathcal{Y} \text{ such that } \forall i \quad f(x_i) = y_i \ . \tag{P}$$

Remarks:

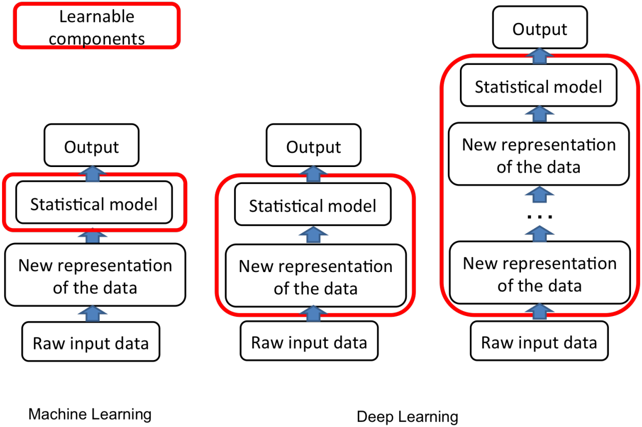
- General setting impossible!
- Solving (P) asks for models on f and an evaluation of the error.

Today's talk

What is it?

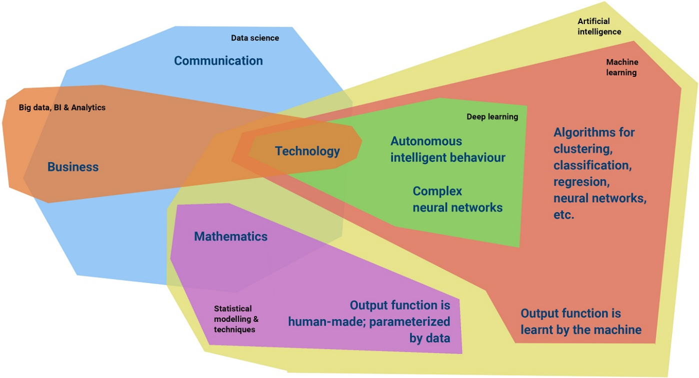

Learnable components
Output
Statistical model
New representation of the data
Raw input data
Machine Learning
Output
Statistical model
New representation of the data
Raw input data
Output
Statistical model
New representation of the data
...
New representation of the data
Raw input data
Deep Learning

What is it?



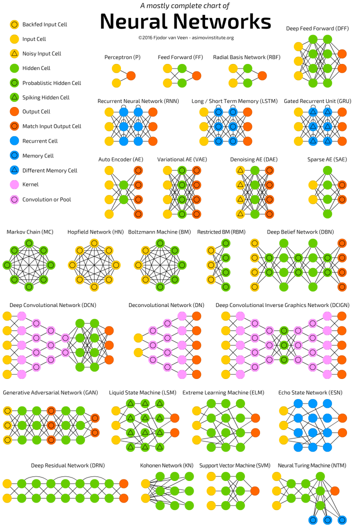
Deep learning ⇒ hierarchical learning with high order features.

The location of deep learning



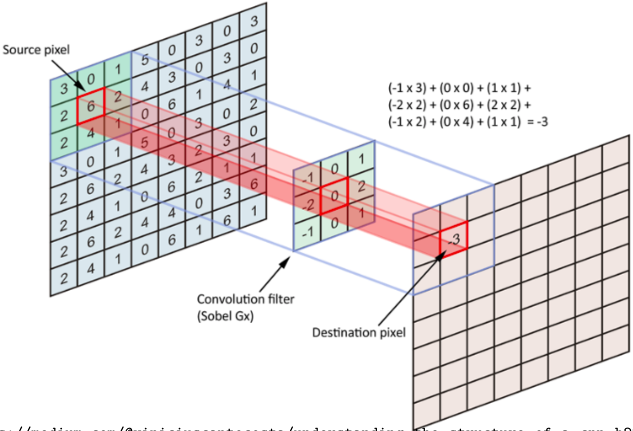
From https://www.machinecurve.com/index.php/2017/09/30/the-differences-between-artificial-intelligence-machine-learning-more/

The zoo



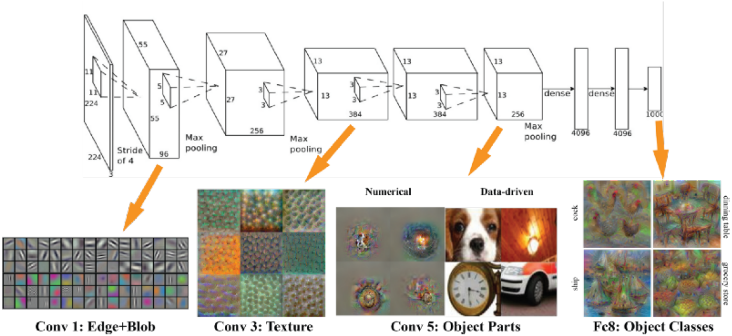
From `http://www.asimovinstitute.org/neural-network-zoo/`

About convolution neural networks



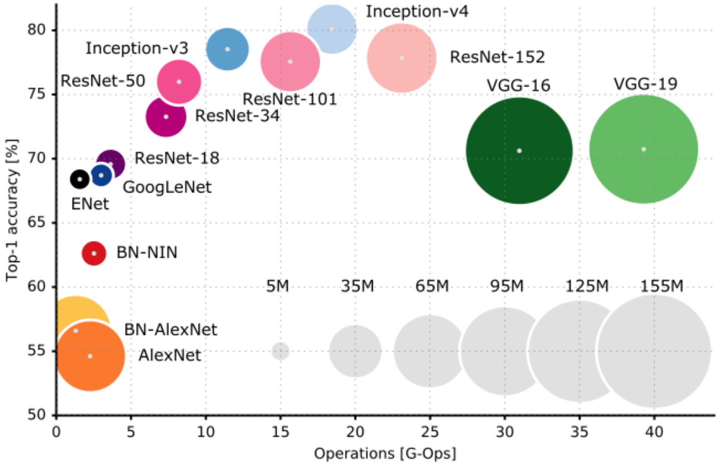
From https://medium.com/@viniciuscantocosta/understanding-the-structure-of-a-cnn-b220148e2ac4

AlexNet



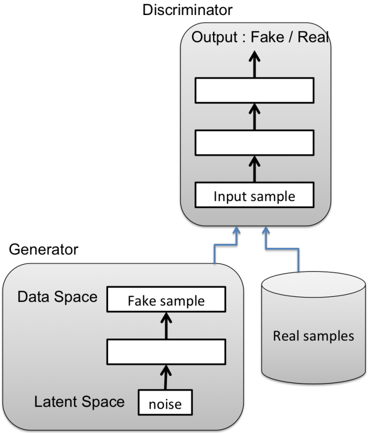
ImageNet Classification with Deep Convolutional Neural Networks by A. Krizhevsky et al (NIPS 2012)

AlexNet (results)



Generative Adversarial Networks (GAN)

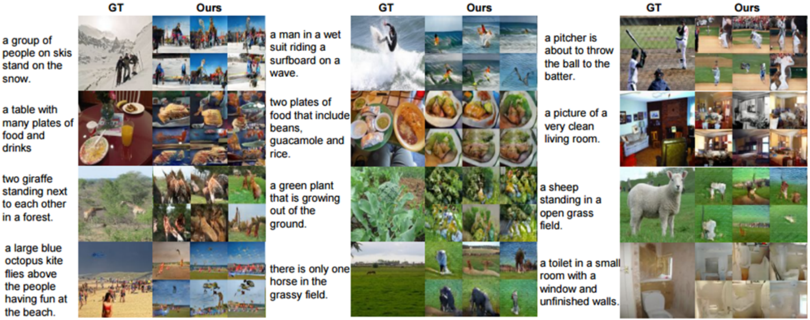
Idea: building a generator that can fool a discriminator



- **Generator**: a NN that produce new data from *noise*.
- **Discriminator**: a classifier which distinguish fake data from true.
- A set of real samples.

Generative adversarial nets by I. Goodfellow et al (NIPS 2014)

Generative Adversarial Networks (GAN): some results



Examples of generated images

From https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-1-Generative-Adversarial-Nets

Source of most data

Today's talk

Image formation setting: AWGN

Using previous setting, we have for $x \in \mathbb{R}^d$,

$$x = y + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$$

Image formation setting: AWGN

Using previous setting, we have for $x \in \mathbb{R}^d$,

$$x = y + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$$

- Ill-posed inverse problem.

Image formation setting: AWGN

Using previous setting, we have for $x \in \mathbb{R}^d$,

$$x = y + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$$

- Ill-posed inverse problem.
- Finding y asks for prior knowledge.

Image formation setting: AWGN

Using previous setting, we have for $x \in \mathbb{R}^d$,

$$x = y + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$$

- Ill-posed inverse problem.
- Finding y asks for prior knowledge.
- σ not always known.

Classical way

General methods: Bayesian way with linear and non-linear methods

Classical way

General methods: Bayesian way with linear and non-linear methods

- Linear filtering.

Classical way

General methods: Bayesian way with linear and non-linear methods

- Linear filtering.
- Ridge regression (e.g smooth prior).

Classical way

General methods: Bayesian way with linear and non-linear methods

- Linear filtering.
- Ridge regression (e.g smooth prior).
- LASSO (e.g sparse prior, TV).

Classical way

General methods: Bayesian way with linear and non-linear methods

- Linear filtering.
- Ridge regression (e.g smooth prior).
- LASSO (e.g sparse prior, TV).
- Non-local mean methods.

Classical way

General methods: Bayesian way with linear and non-linear methods

- Linear filtering.
- Ridge regression (e.g smooth prior).
- LASSO (e.g sparse prior, TV).
- Non-local mean methods.
- Collaborative filtering (e.g BM3D)...

Classical way: the drawback

Priors are rarely completely appropriate,

Classical way: the drawback

Priors are rarely completely appropriate,

- Most images are rather **sharp** than smooth (e.g edges).

Classical way: the drawback

Priors are rarely completely appropriate,

- Most images are rather **sharp** than smooth (e.g edges).
- Sparsity asks for an appropriate **dictionary**.

Classical way: the drawback

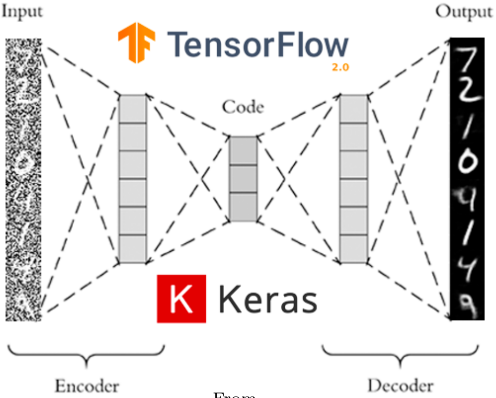
Priors are rarely completely appropriate,

- Most images are rather **sharp** than smooth (e.g edges).
- Sparsity asks for an appropriate **dictionary**.
- Non-local mean and collaborative filtering assume **redundancy** inside the image.

The deep learning way

Idea: learn the prior, through representation leaning, from a database of clean images

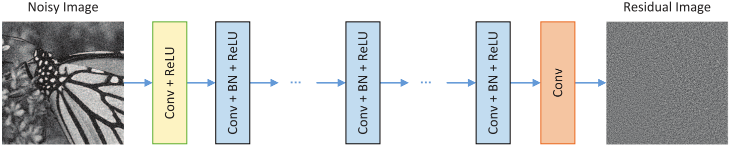
Example: Denoising autoencoder [Vincent et al, IMCL 2008]



From https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/

DnCNN: the first deep way

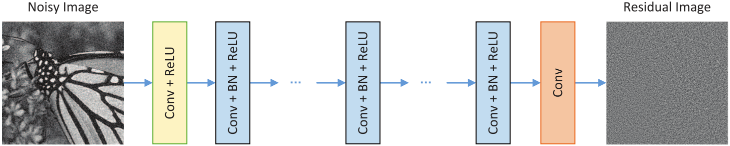
Idea: learn to remove the noise with a fully convolutional deep neural network by learning the noise!



From *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising* [Zhang et al, IEEE TIP 2016]

DnCNN: the first deep way

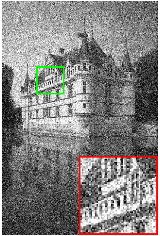
Idea: learn to remove the noise with a fully convolutional deep neural network by learning the noise!



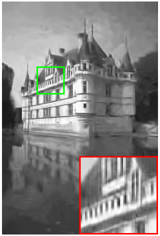
From *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising* [Zhang et al, IEEE TIP 2016]

Remark: The other way, learning to directly denoise, also works.

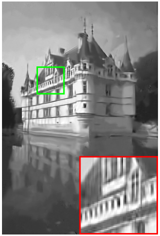
Some results (first part)



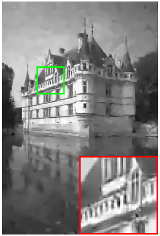
(a) Noisy / 14.76dB



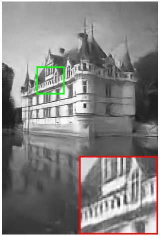
(b) BM3D / 26.21dB



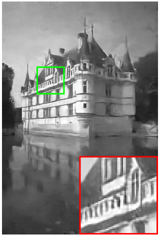
(c) WNNM / 26.51dB



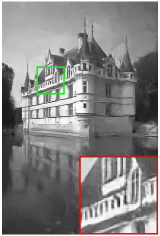
(d) EPLL / 26.36dB



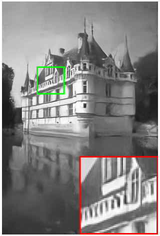
(e) MLP / 26.54dB



(f) TNRD / 26.59dB



(g) DnCNN-S / 26.90dB



(h) DnCNN-B / 26.92dB

From [Zhang et al, IEEE TIP 2016]

Playing with the training process aka Noise2noise

Let's recall the image formation,

$$x = y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

Idea: if $\mathbb{E}(\varepsilon) = 0$, then $\mathbb{E}(y|x) = y$.
Principle at the learning step:

Playing with the training process aka Noise2noise

Let's recall the image formation,

$$x = y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

Idea: if $\mathbb{E}(\varepsilon) = 0$, then $\mathbb{E}(y|x) = y$.
Principle at the learning step:

- Take a noise distribution with zero mean.

Playing with the training process aka Noise2noise

Let's recall the image formation,

$$x = y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

Idea: if $\mathbb{E}(\varepsilon) = 0$, then $\mathbb{E}(y|x) = y$.
Principle at the learning step:

- Take a noise distribution with zero mean.
- Generating new noisy image at each steps.

Playing with the training process aka Noise2noise

Let's recall the image formation,

$$x = y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

Idea: if $\mathbb{E}(\varepsilon) = 0$, then $\mathbb{E}(y|x) = y$.
Principle at the learning step:

- Take a noise distribution with zero mean.
- Generating new noisy image at each steps.
- The target is also another noisy image.

Playing with the training process aka Noise2noise

Let's recall the image formation,

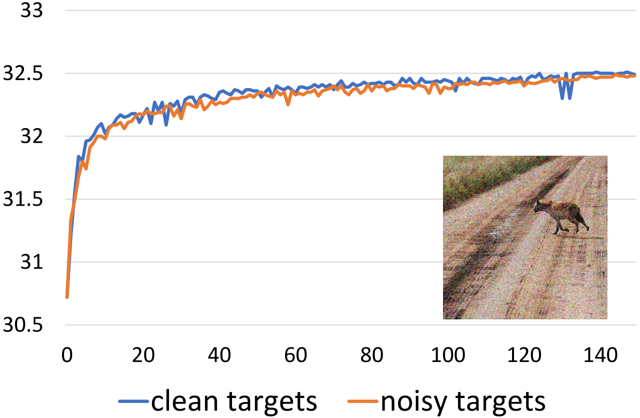
$$x = y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

Idea: if $\mathbb{E}(\varepsilon) = 0$, then $\mathbb{E}(y|x) = y$.
Principle at the learning step:

- Take a noise distribution with zero mean.
- Generating new noisy image at each steps.
- The target is also another noisy image.

Better than previous way aka noise2clean!

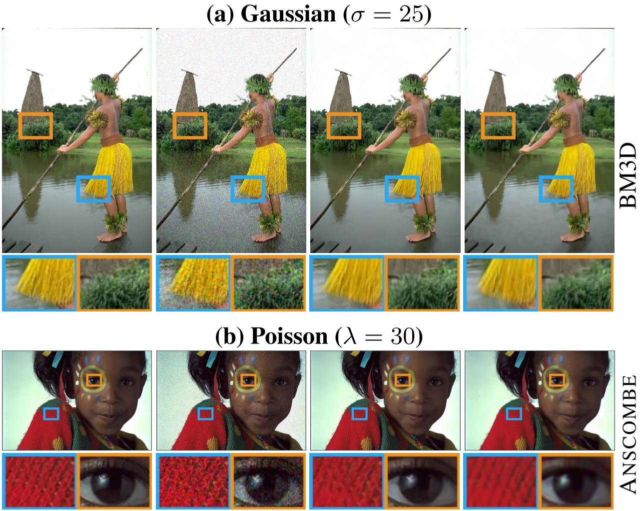
Some results: n2c vs n2n



(a) White Gaussian, $\sigma = 25$

From [Lehtinen et al, ICML 2018]

Some results (second part)



From [Lehtinen et al, ICML 2018]

Toward non-supervised ways: Noise2Void

Remark: previous methods ask for a **clean** database $\Rightarrow$ not always available

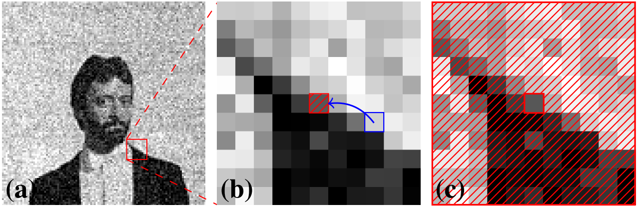
Toward non-supervised ways: Noise2Void

Remark: previous methods ask for a **clean** database $\Rightarrow$ not always available
Idea: learning from noisy images by trying to estimate a missing pixel

Toward non-supervised ways: Noise2Void

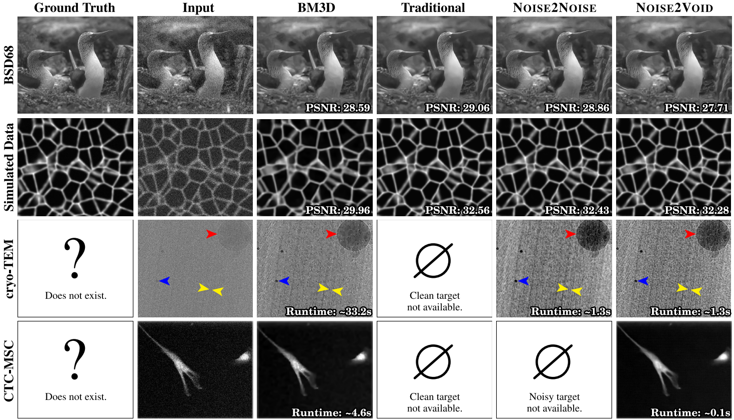
Remark: previous methods ask for a **clean** database ⇒ not always available

Idea: learning from noisy images by trying to estimate a missing pixel



From [Krull et al, ArXiv:1811.10980]

Some results (third part)



From [Krull et al, ArXiv:1811.10980]

Dealing with other noises

Adapting the loss function to deal with other noises,

Dealing with other noises

Adapting the loss function to deal with other noises,

- ℓ_2-norm (or MSE) for Gaussian noise.

Dealing with other noises

Adapting the loss function to deal with other noises,

- ℓ_2-norm (or MSE) for Gaussian noise.
- ℓ_1-norm for dealing with salt and pepper noise.

Dealing with other noises

Adapting the loss function to deal with other noises,

- ℓ_2-norm (or MSE) for Gaussian noise.
- ℓ_1-norm for dealing with salt and pepper noise.
- Using MSE and VST (e.g Anscombe) for Poisson noise.

Dealing with other noises

Adapting the loss function to deal with other noises,

- ℓ_2-norm (or MSE) for Gaussian noise.
- ℓ_1-norm for dealing with salt and pepper noise.
- Using MSE and VST (e.g Anscombe) for Poisson noise.

The universal denoiser still an open question.

Summing up

The step for to build a denoiser with deep learning methods,

Summing up

The step for to build a denoiser with deep learning methods,

1. Select a fully convolutional neural network (e.g DnCNN, UNet).

Summing up

The step for to build a denoiser with deep learning methods,

1. Select a fully convolutional neural network (e.g DnCNN, UNet).
2. Build a clean or noisy database.

Summing up

The step for to build a denoiser with deep learning methods,

1. Select a fully convolutional neural network (e.g DnCNN, UNet).
2. Build a clean or noisy database.
3. Select the appropriate loss function (depends on the noise distribution).

Summing up

The step for to build a denoiser with deep learning methods,

1. Select a fully convolutional neural network (e.g DnCNN, UNet).
2. Build a clean or noisy database.
3. Select the appropriate loss function (depends on the noise distribution).
4. Select a learning procedure.

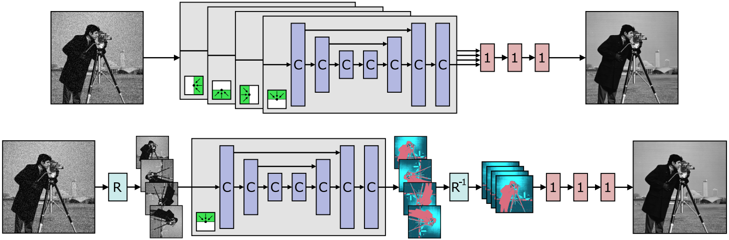
Summing up

The step for to build a denoiser with deep learning methods,

1. Select a fully convolutional neural network (e.g DnCNN, UNet).
2. Build a clean or noisy database.
3. Select the appropriate loss function (depends on the noise distribution).
4. Select a learning procedure.
5. Enjoy.

Recent improvements

Toward higher quality by [Laine et al, NeurIPS 2019]



Today's talk

What about deblurring?

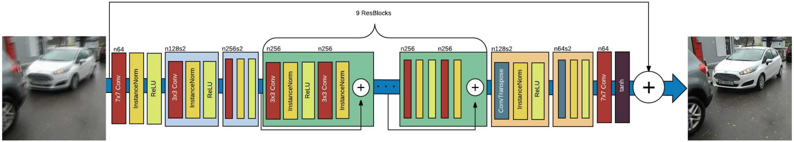
Take a more complex image formation formula,

$$x = h \circ y + \varepsilon, \quad \varepsilon \text{ follow some distributions.}$$

- $\circ$ the convolution operator.
- h the blur kernel.

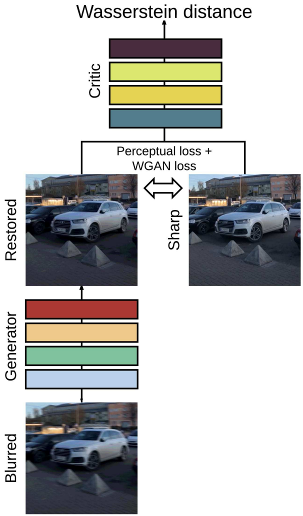
Toward fully deep neural networks

DeblurGAN [Kupyn et al, CVPR 2018]: Mixing GAN with residual blocks.



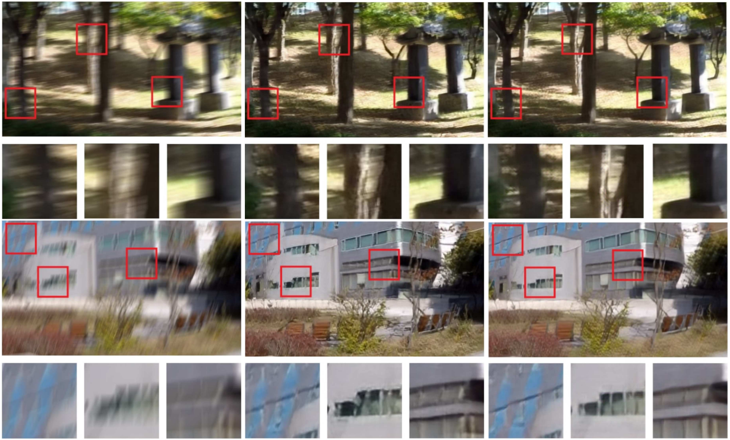
From [Kupyn et al, CVPR 2018]

Learning using GAN



From [Kupyn et al, CVPR 2018]

Some deblurring results



Results on the GoPro test dataset. From left to right: blurred photo, Nah *et al.* [25], DeblurGAN.

From [Kupyn et al, CVPR 2018]

Some words of warning

1. Neural networks can be ineffective for some inverse problems [Chodosh and Lucey, CVPR 2020].

Some words of warning

1. Neural networks can be ineffective for some inverse problems [Chodosh and Lucey, CVPR 2020].
2. The database must be mostly “complete”.

Some words of warning

1. Neural networks can be ineffective for some inverse problems [Chodosh and Lucey, CVPR 2020].

2. The database must be mostly “complete”.

3. Can be unstable! [Gottschling et al, 2020; Genzel et al, 2020].

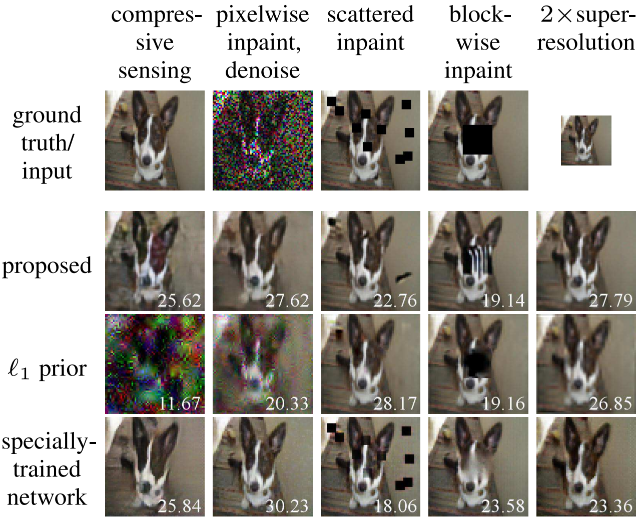
One way to solve them all

Idea [Chang et al, ICCV 2017]: using the deep denoising network as a projector and then solve

$$\min_y \tfrac{1}{2}\|\mathrm{A}y - x\|^2 + \lambda\phi(x)$$

- A a linear operator (e.g convolution with kernel h, mask operator).
- ϕ an indicatrice function of a set whose projector is given by the neural network.

Some results



From [Chang et al, ICCV 2017]

Take away message

- Evaluation is primordial! Must be coherent with the task.

Take away message

- Evaluation is primordial! Must be coherent with the task.

- Learning representation is a way to avoid some bias.

Take away message

- Evaluation is primordial! Must be coherent with the task.
- Learning representation is a way to avoid some bias.
- Deep learning asks for big datasets, but scale very well.

Take away message

- Evaluation is primordial! Must be coherent with the task.
- Learning representation is a way to avoid some bias.
- Deep learning asks for big datasets, but scale very well.
- There is no free lunch!

Thank you for your attention.

Any questions?