Artificial “Neural Networks”

The questions to ask

Jean Feydy

Harvey Cushing symposium, Paris – June 11^{th}, 2019

Écoles Normales Supérieures de Paris et Paris-Saclay

About me

Jean Feydy (2016-2019) :

- PhD student with Alain Trouvé, ENS Cachan,
  Computational anatomy.
- Teaching assistant for Gabriel Peyré, M2 MVA and ENS Paris,
  Mathematical foundations of data sciences.
- Son of Antoine Feydy, PUPH at the Paris-Cochin hospital,
  Specialist of **musculoskeletal imaging.**

Today

Neural networks?

Today

Neural networks?

A generalization of **linear regression**

to complex models.

Today

Neural networks?

A generalization of **linear regression**
to complex models.

Does it work? Usually, **no, it doesn't**.

Today

Neural networks?

A generalization of **linear regression**
to complex models.

Does it work? Usually, **no, it doesn't**.

However, this idea now allows us to implement
excellent **feature detectors**.

Today

Neural networks?

A generalization of **linear regression**
to complex models.

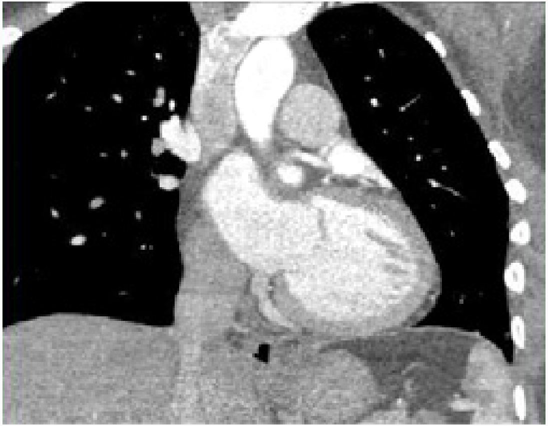
Does it work? Usually, **no, it doesn't**.

However, this idea now allows us to implement
excellent **feature detectors**.

As physicians, how can you look at these
methods with a **critical eye**?

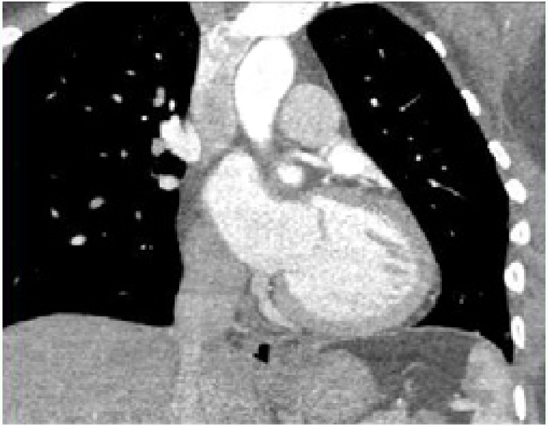
What can we see on a medical image?

One image : three levels of analysis [EPW11, Man11]



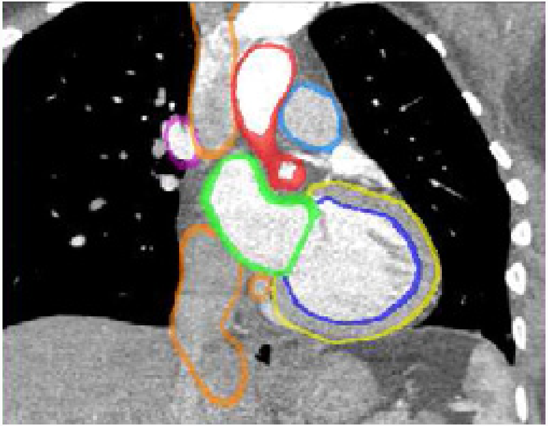
One image : three levels of analysis [EPW11, Man11]

1. Texture



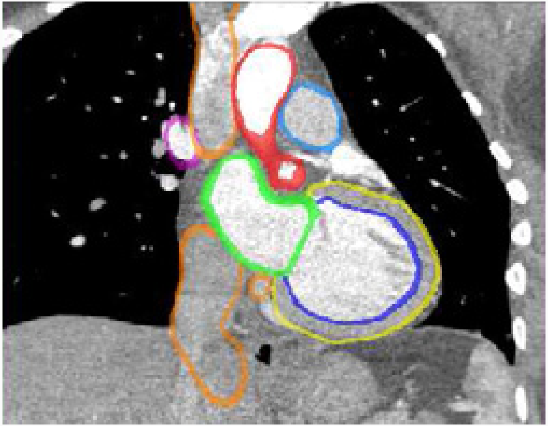
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy



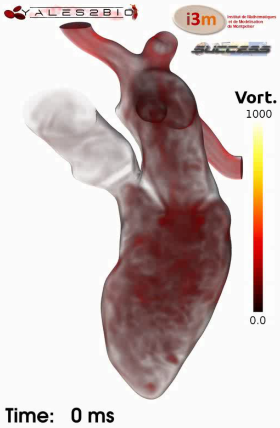
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy



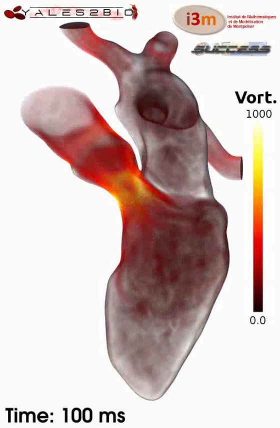
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



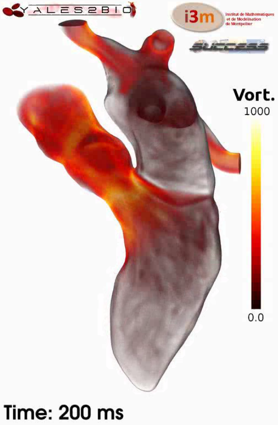
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



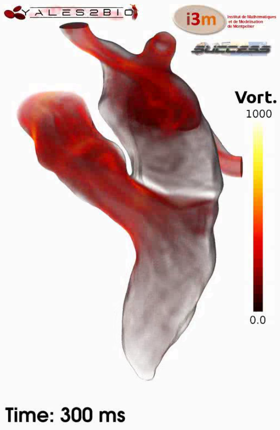
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



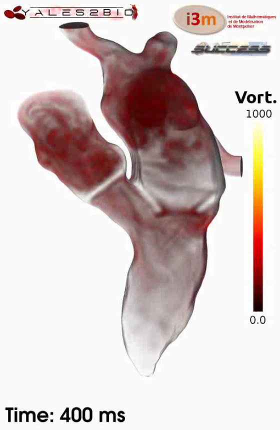
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



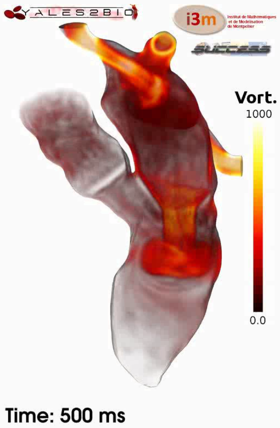
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



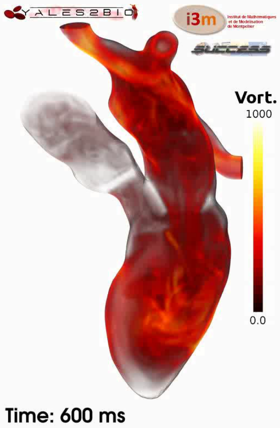
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



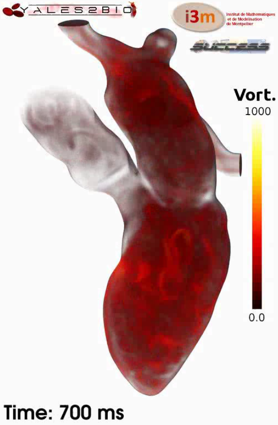
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



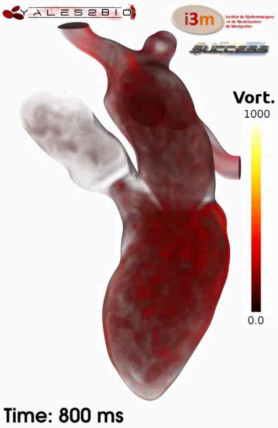
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



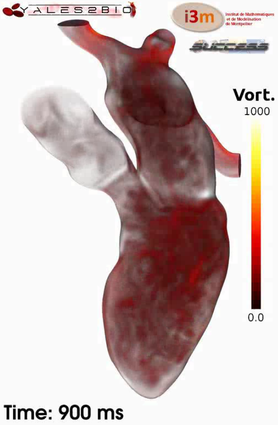
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



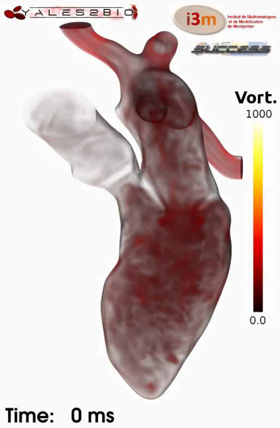
One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function



One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function

Each level of analysis can be **modeled** by relying on the previous one.

One image : three levels of analysis [EPW11, Man11]

1. Texture
2. Anatomy
3. Function

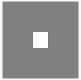
Each level of analysis can be **modeled** by relying on the previous one.

Let's discover the most fundamental of all imaging theories: Texture analysis through **multi-scale filtering**.

Filtering, aka. “convolution product”

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “$a \cdot x$”,
parameterized by the coefficients of a **small filter** φ.



$\star$

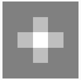
$=$

φ $\qquad$ x $\qquad$ $\varphi \star x$

Filtering, aka. "convolution product"

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "$a \cdot x$",
parameterized by the coefficients of a **small filter** φ.

 $\star$  $=$ 

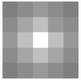
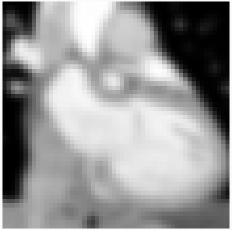
φ $\qquad$ x $\qquad$ $\varphi \star x$

Filtering, aka. "convolution product"

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.

 $\star$  $=$ 

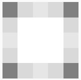
φ $\qquad$ x $\qquad$ $\varphi \star x$

Filtering, aka. "convolution product"

Convolution (i.e. weighted average of the neighboring pixels) :

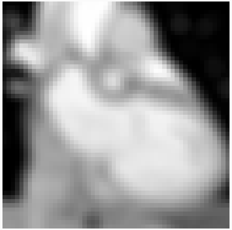
Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.



$\star$

$=$



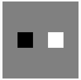
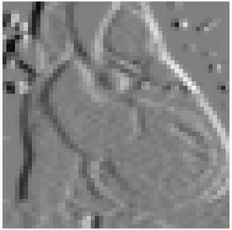
φ $\quad$ x $\quad$ $\varphi \star x$

Filtering, aka. “convolution product”

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “$a \cdot x$”,

parameterized by the coefficients of a **small filter** φ.

 $\star$  $=$ 

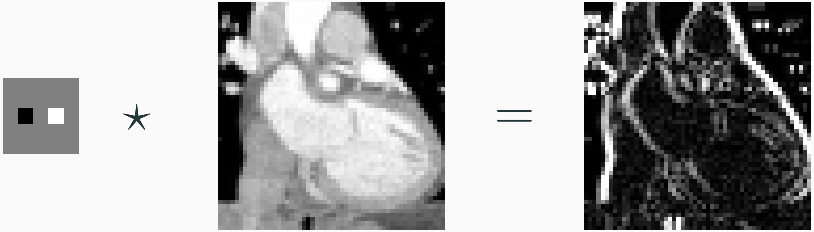
φ $\qquad$ x $\qquad$ $\varphi \star x$

Filtering, aka. “convolution product”

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “$a \cdot x$”,

parameterized by the coefficients of a **small filter** φ.



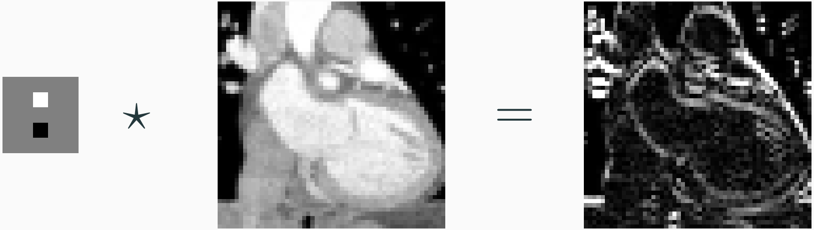
$\varphi \qquad\qquad x \qquad\qquad |\varphi \star x|$

Filtering, aka. "convolution product"

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.

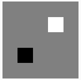


φ $\qquad$ x $\qquad$ $|\varphi \star x|$

Filtering, aka. “convolution product”

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “$a \cdot x$”,
parameterized by the coefficients of a **small filter** φ.



$\star$

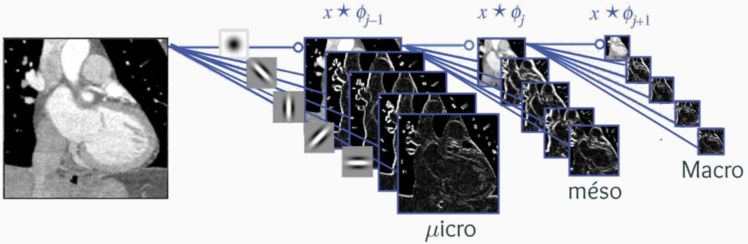
$=$

φ $\qquad$ x $\qquad$ $|\varphi \star x|$

Multi-scale prior on images

Wavelet theory (1990∼2010 ; Meyer, Mallat, Daubechies...) :

Small filters + cascading zoom-out operations [Mal16]:



Multi-scale prior on images

Wavelet theory (1990∼2010 ; Meyer, Mallat, Daubechies...) :

Small filters + cascading zoom-out operations [Mal16]:

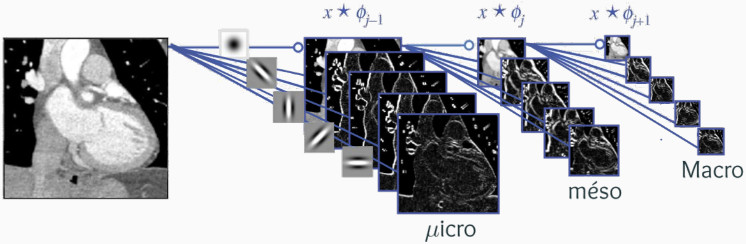


Image	$\longrightarrow$	Relevant coefficients
$\simeq$ ".wav" Audio	$\longrightarrow$	Music score

Multi-scale prior on images

Wavelet theory (1990$\sim$2010 ; Meyer, Mallat, Daubechies...) :

Small filters + cascading zoom-out operations [Mal16]:

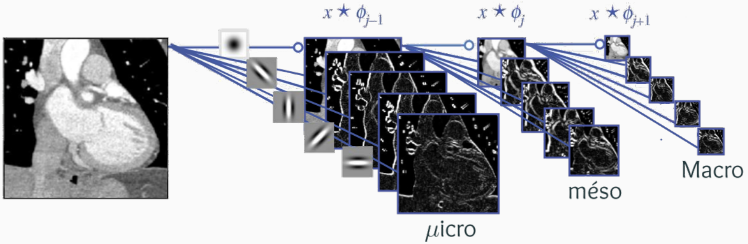


Image	$\longrightarrow$	Relevant coefficients
$\simeq$ ".wav" Audio	$\longrightarrow$	Music score

$\Longrightarrow$ **JPEG2000** format, standard of the movie industry.

That's it for classical models.
What about neural networks?

“Supervised learning” = regression

We have:

- A database $\{(x_1 \to y_1), (x_2 \to y_2), \ldots\}$.

"Supervised learning" = regression

We have:

- A database $\{(x_1 \to y_1), (x_2 \to y_2), \ldots\}$.
- A model

$$F(\underbrace{\mathbf{w}}_{\text{parameters}} ; \underbrace{x}_{\text{input}}) \to \underbrace{y}_{\text{output}},$$

“Supervised learning” = regression

We have:

- A database $\{(x_1 \to y_1), (x_2 \to y_2), \ldots\}$.
- A model

$$\underbrace{F(\mathbf{w}}_{\text{parameters}}\,;\,\underbrace{x}_{\text{input}}) \to \underbrace{y}_{\text{output}},$$

“Supervised learning” = regression

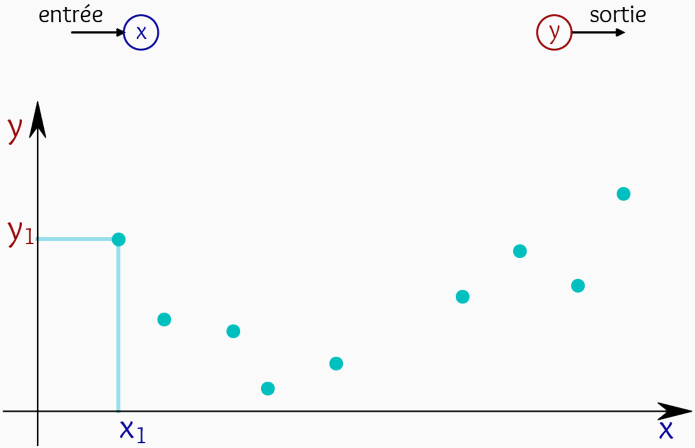
We have:

- A database $\{(x_1 \to y_1), (x_2 \to y_2), \dots\}$.
- A model

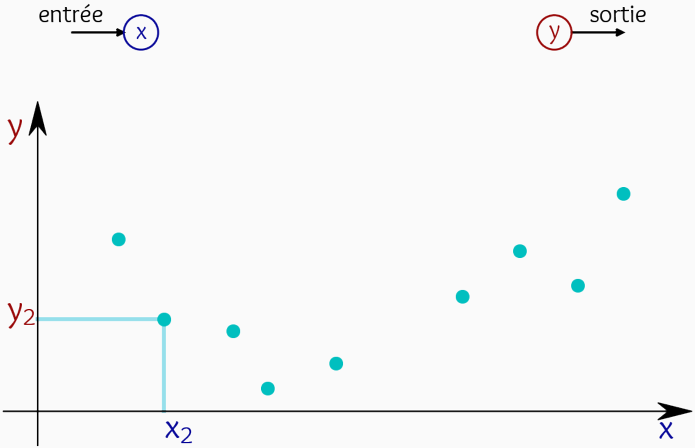
$$\underbrace{F(\mathbf{w}}_{\text{parameters}} ; \underbrace{x}_{\text{input}}) \to \underbrace{y}_{\text{output}},$$

Let’s find, step by step, a value $\mathbf{w}_{\text{optimal}}$ of the parameters that minimizes **the average error on the predictions.**

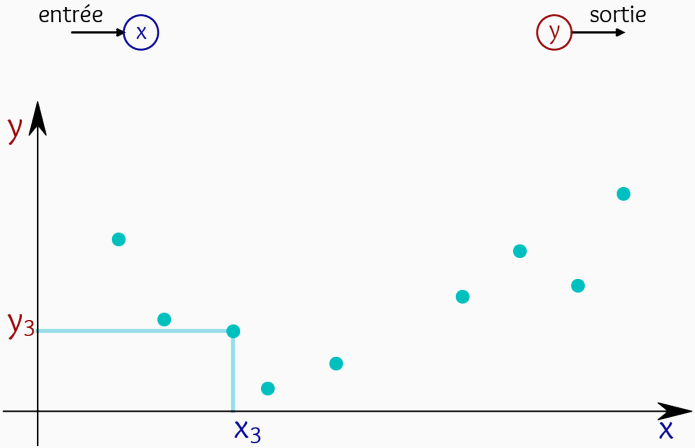
A toy dataset, in dimension 1



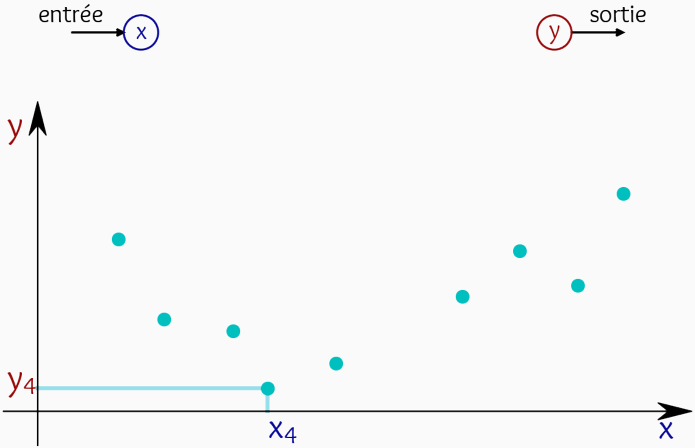
A toy dataset, in dimension 1



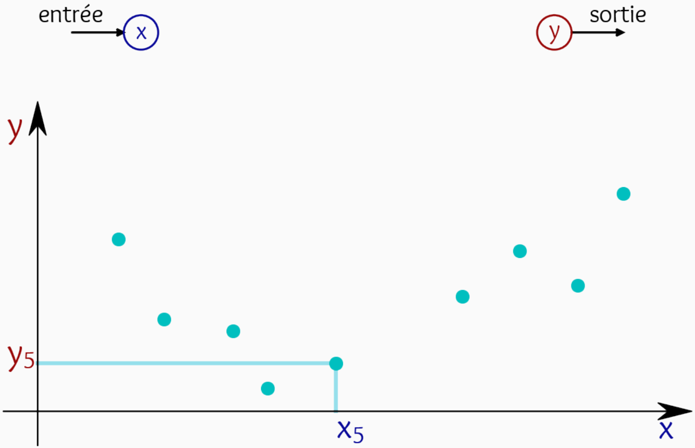
A toy dataset, in dimension 1



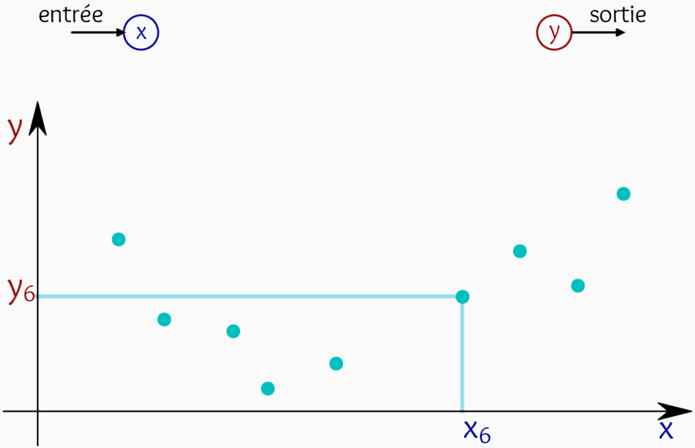
A toy dataset, in dimension 1



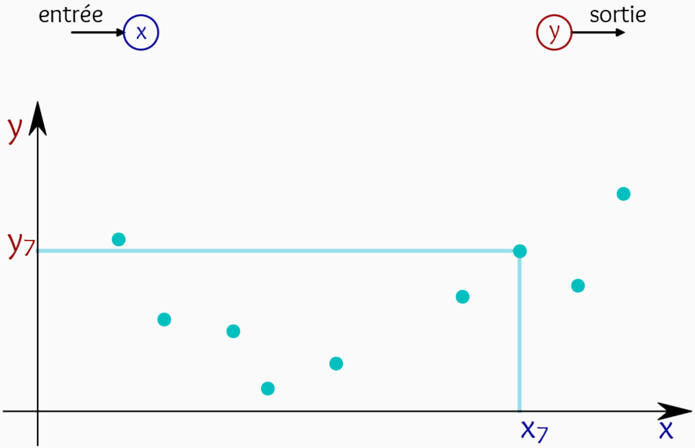
A toy dataset, in dimension 1



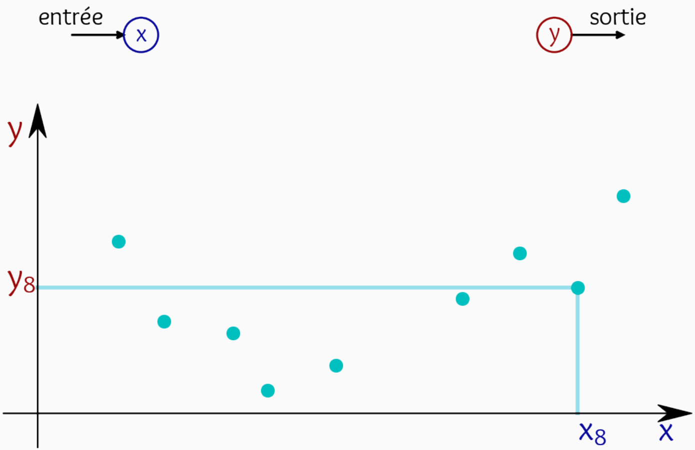
A toy dataset, in dimension 1



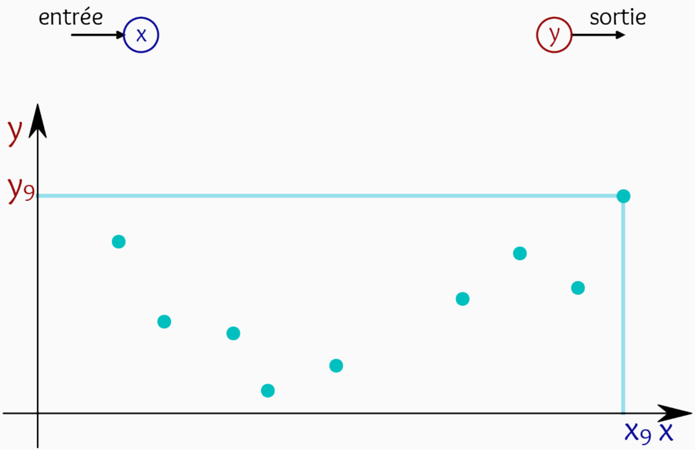
A toy dataset, in dimension 1



A toy dataset, in dimension 1

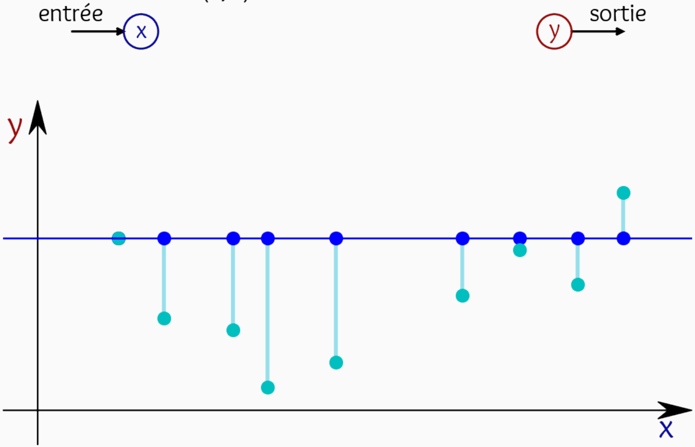


A toy dataset, in dimension 1



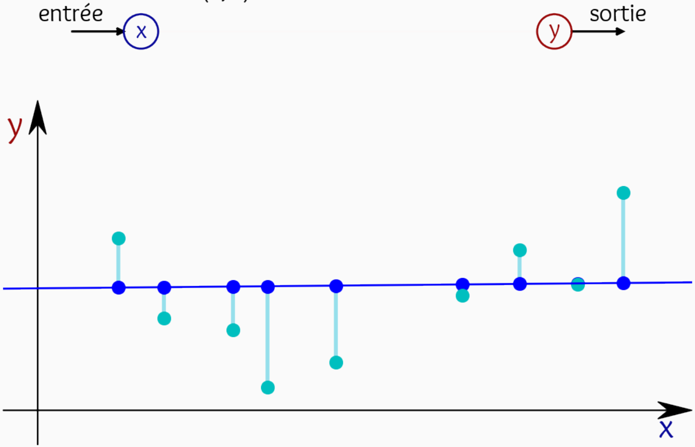
A fundamental example: linear regression

$$F(a, b; x) = a \cdot x + b$$
$$(a, b) = +0.00,\ +0.25$$



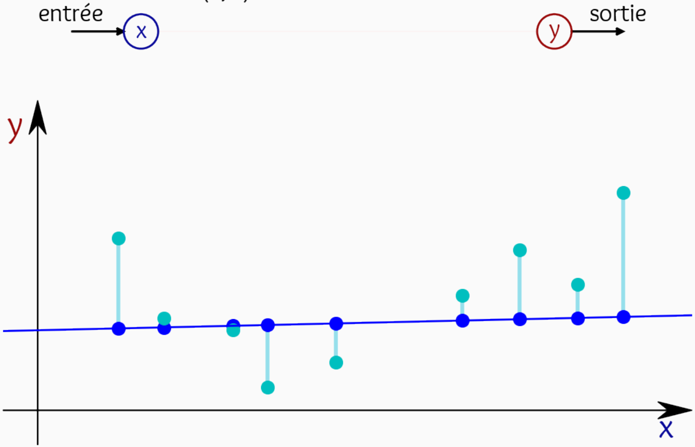
A fundamental example: linear regression

$$F(a,b;x) = a \cdot x + b$$
$$(a,b) = +0.01,\ +0.18$$

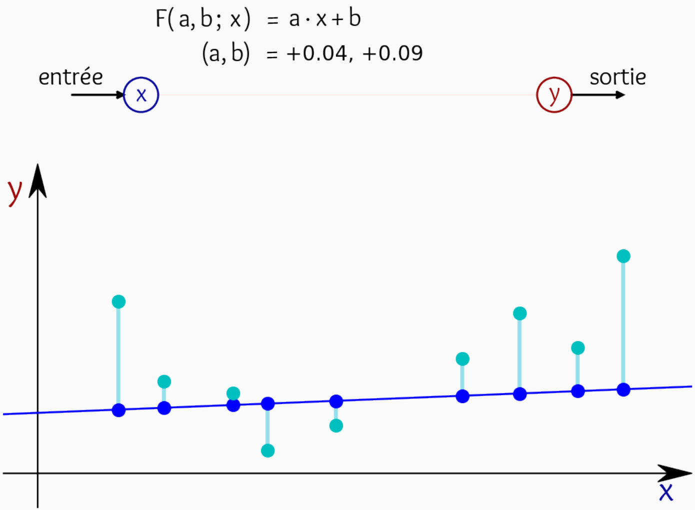


A fundamental example: linear regression

$$F(a,b;x) = a \cdot x + b$$
$$(a,b) = +0.02,\ +0.12$$

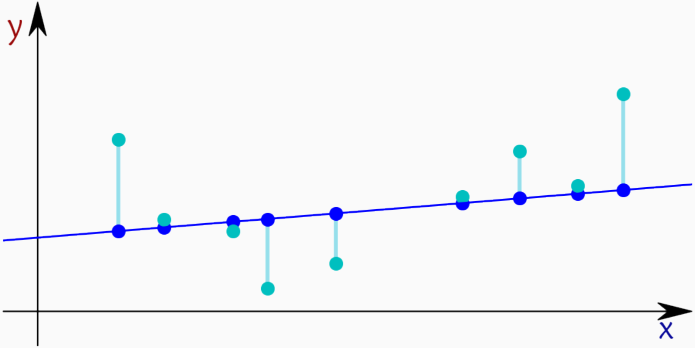


A fundamental example: linear regression



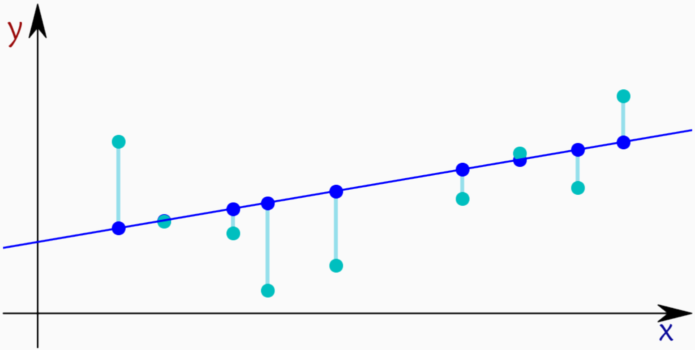
A fundamental example: linear regression

$$F(a, b; x) = a \cdot x + b$$
$$(a, b) = +0.08,\ +0.11$$



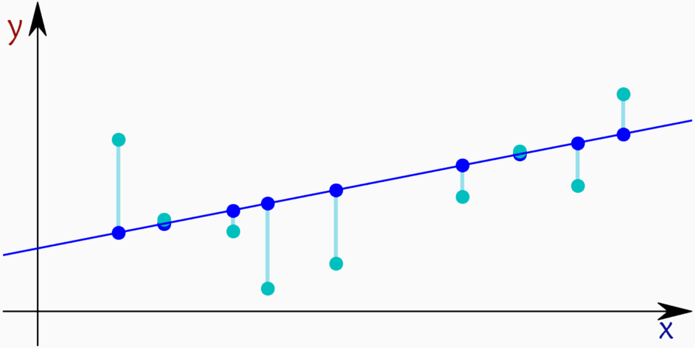
A fundamental example: linear regression

$$F(a, b; x) = a \cdot x + b$$
$$(a, b) = +0.17,\ +0.10$$



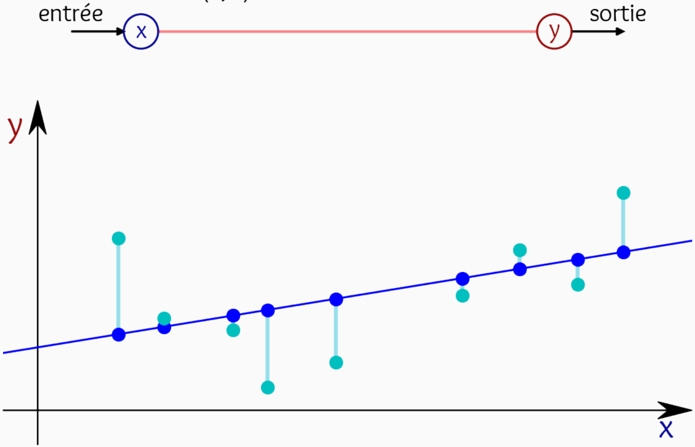
A fundamental example: linear regression

$$F(a,b;x) = a \cdot x + b$$
$$(a,b) = +0.20,\ +0.09$$



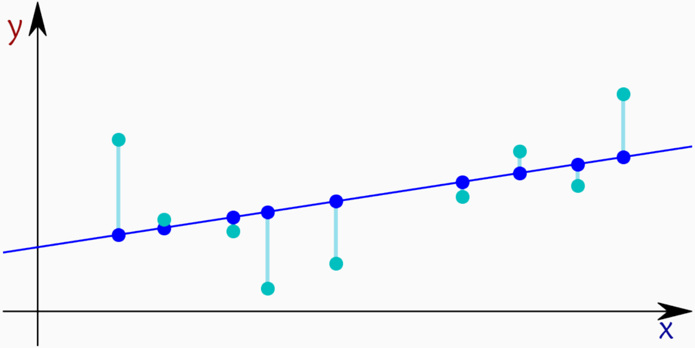
A fundamental example: linear regression

$$F(a, b; x) = a \cdot x + b$$
$$(a, b) = +0.16,\ +0.09$$

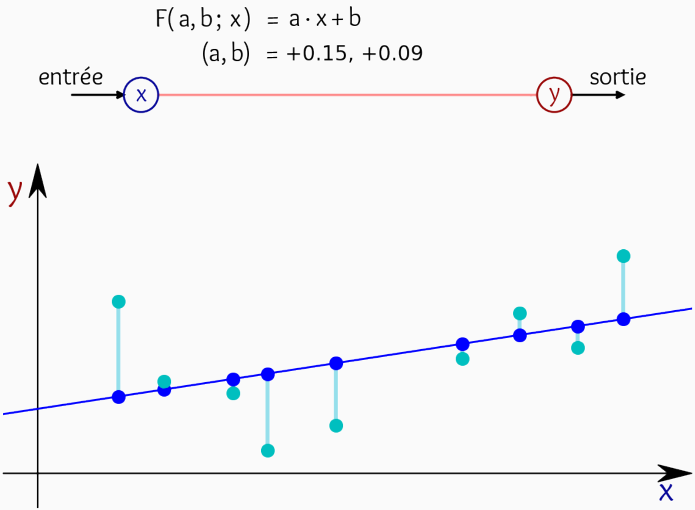


A fundamental example: linear regression

$$F(a, b; x) = a \cdot x + b$$
$$(a, b) = +0.15, +0.09$$



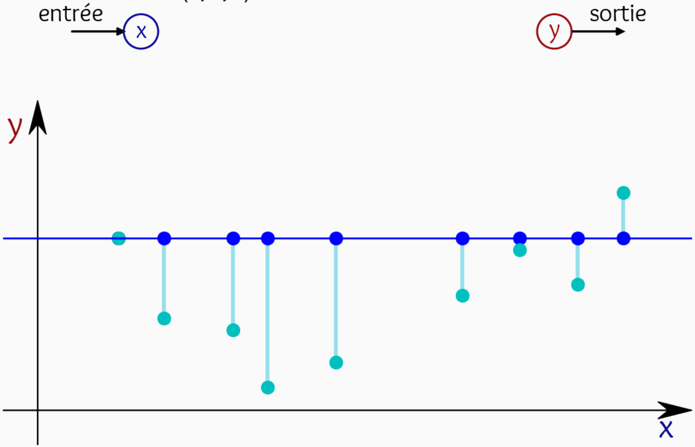
A fundamental example: linear regression



A slightly more complex model: quadratic regression

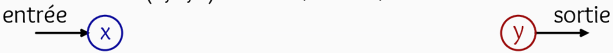
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

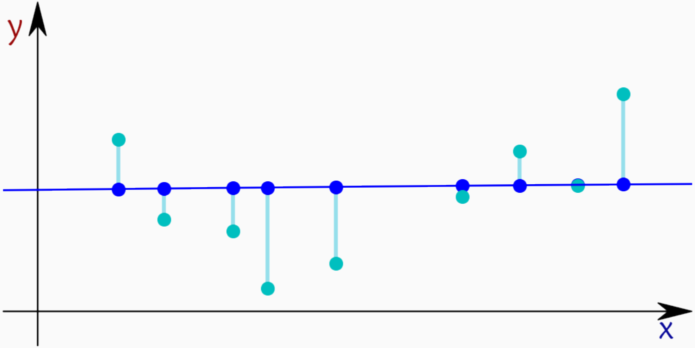
$$(a, b, c) = +0.00,\ +0.00,\ +0.25$$



A slightly more complex model: quadratic regression

$$F(a,b,c;x) = a \cdot x^2 + b \cdot x + c$$
$$(a,b,c) = -0.00,\ +0.01,\ +0.18$$

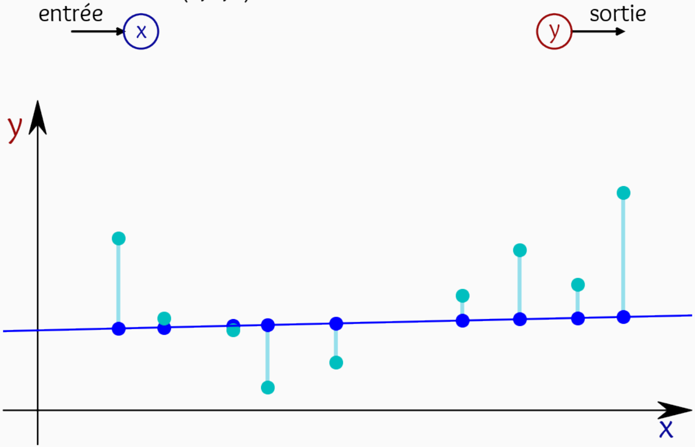




A slightly more complex model: quadratic regression

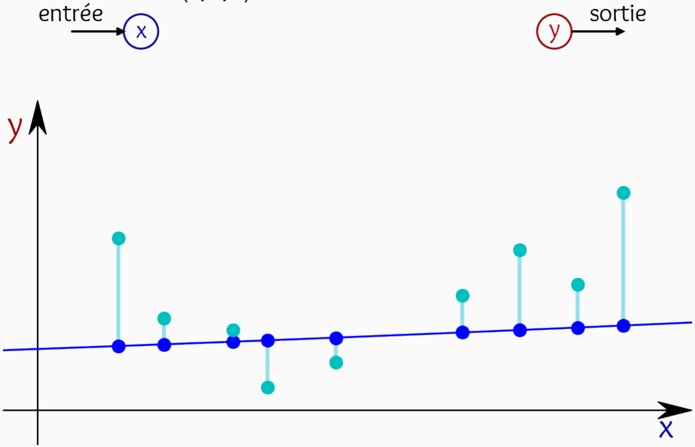
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = -0.00,\ +0.02,\ +0.12$$



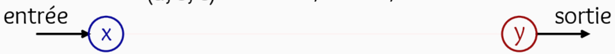
A slightly more complex model: quadratic regression

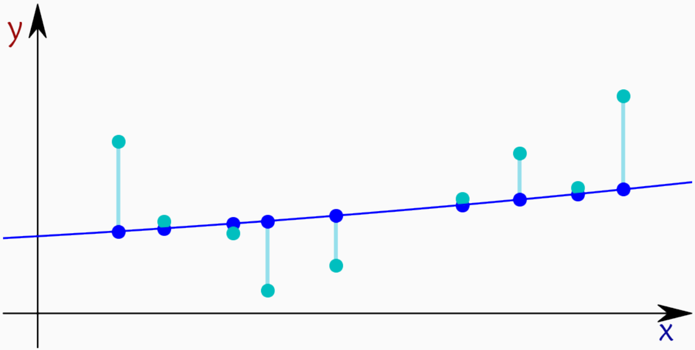
$$F(a,b,c;x) = a \cdot x^2 + b \cdot x + c$$
$$(a,b,c) = +0.00,\ +0.04,\ +0.09$$



A slightly more complex model: quadratic regression

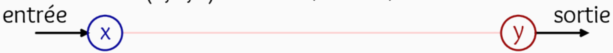
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +0.03,\ +0.08,\ +0.11$$

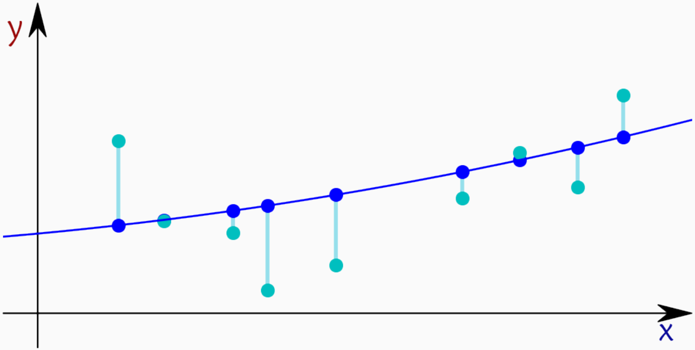




A slightly more complex model: quadratic regression

$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +0.08,\ +0.17,\ +0.12$$

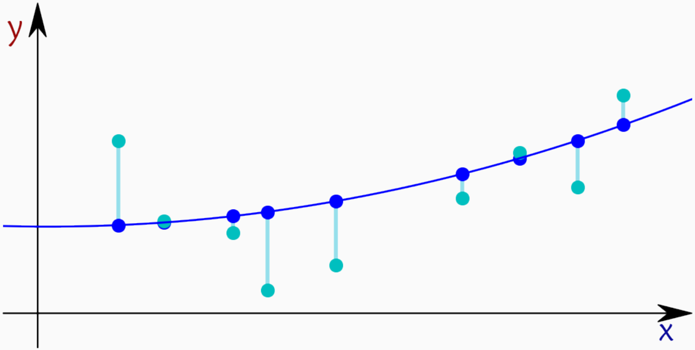




A slightly more complex model: quadratic regression

$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

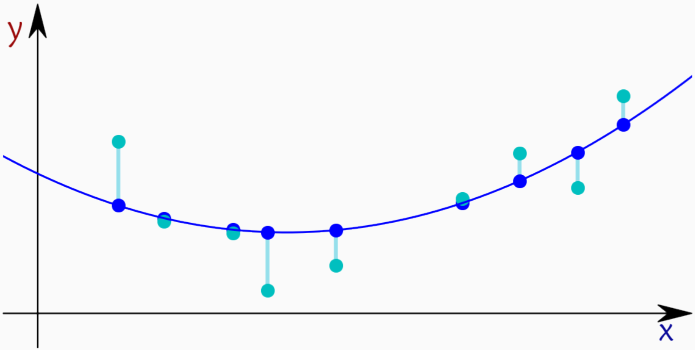
$$(a, b, c) = +0.21,\ +0.18,\ +0.13$$



A slightly more complex model: quadratic regression

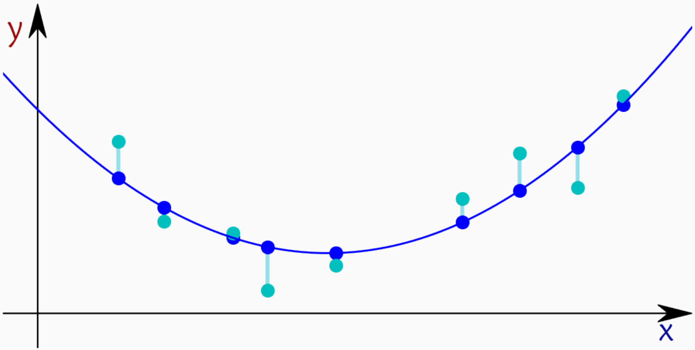
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +0.66,\ +0.12,\ +0.20$$



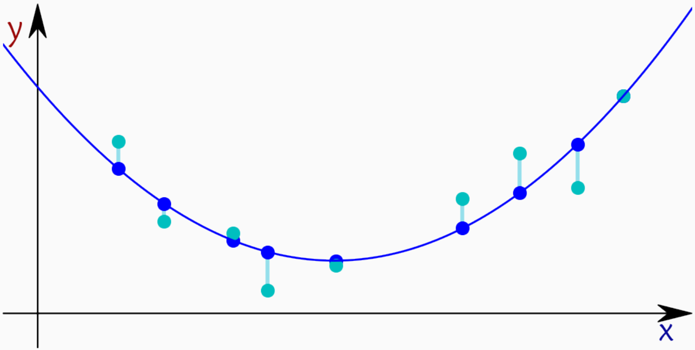
A slightly more complex model: quadratic regression

$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +1.18,\ +0.07,\ +0.30$$



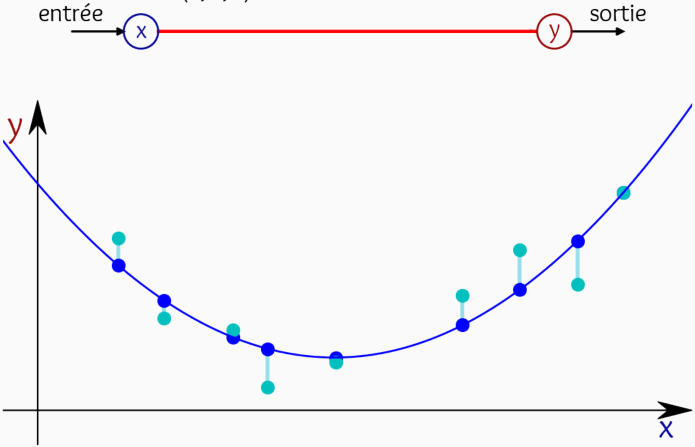
A slightly more complex model: quadratic regression

$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +1.36,\ +0.05,\ +0.33$$



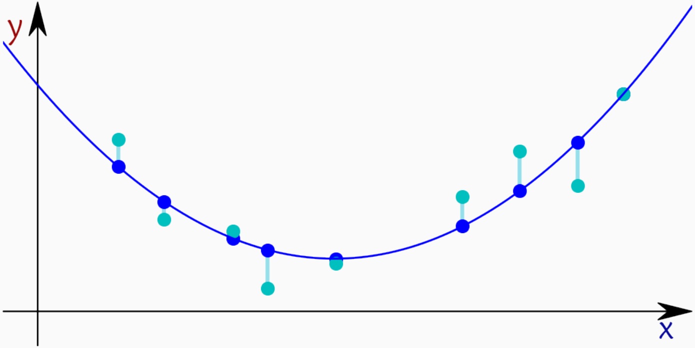
A slightly more complex model: quadratic regression

$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +1.36,\ +0.05,\ +0.33$$

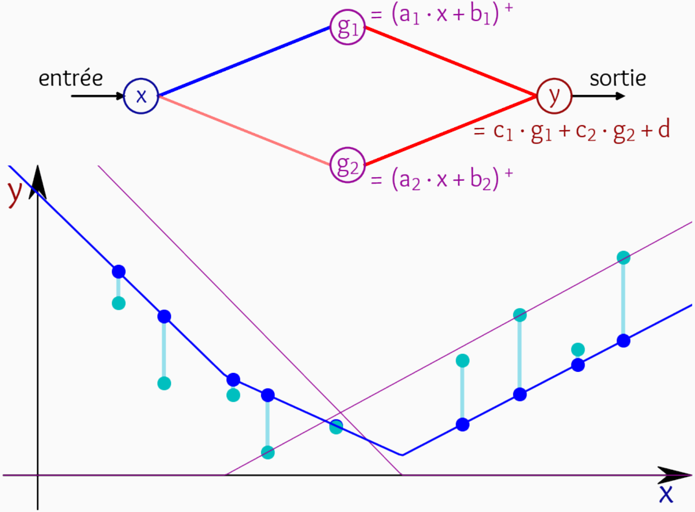


A slightly more complex model: quadratic regression

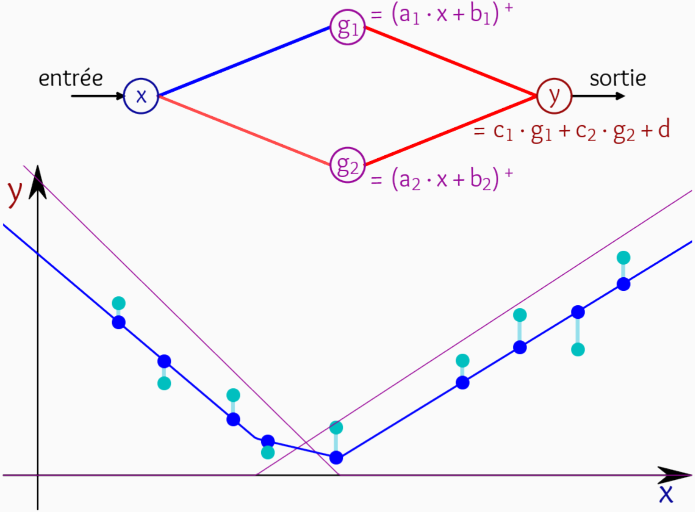
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$
$$(a, b, c) = +1.36,\ +0.05,\ +0.33$$



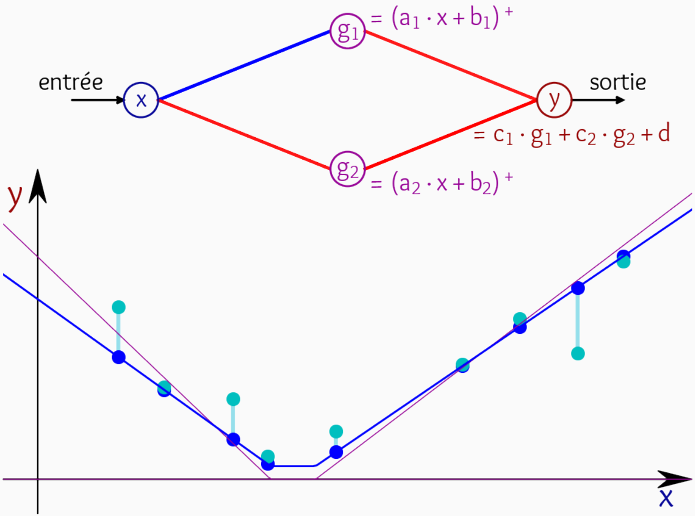
Let's complexify our model with intermediate variables...



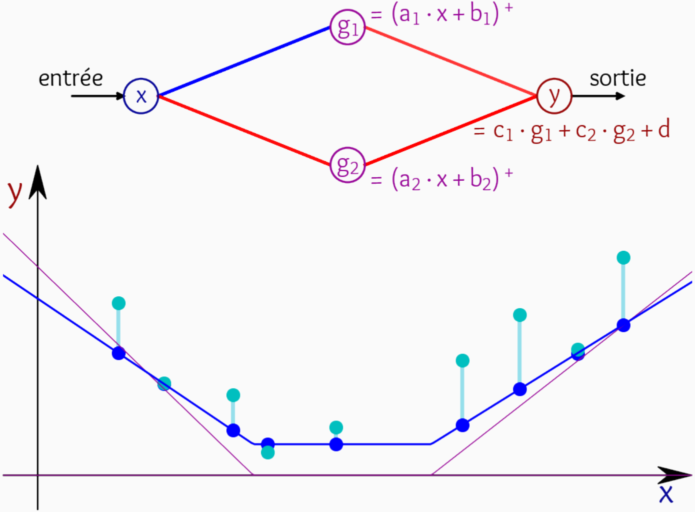
Let's complexify our model with intermediate variables...



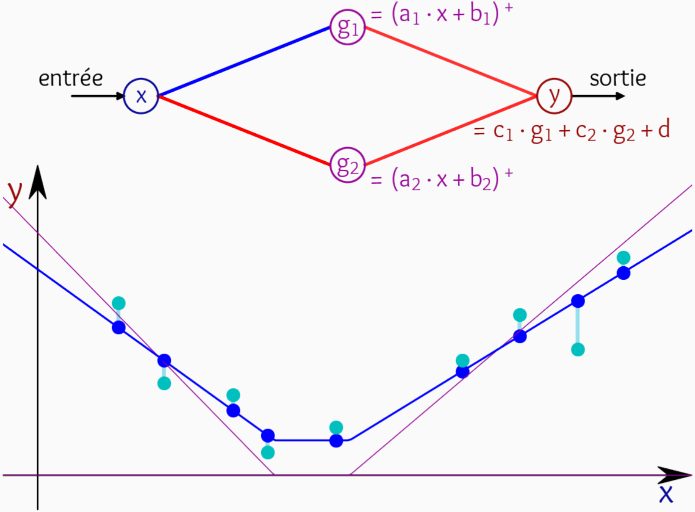
Let's complexify our model with intermediate variables...



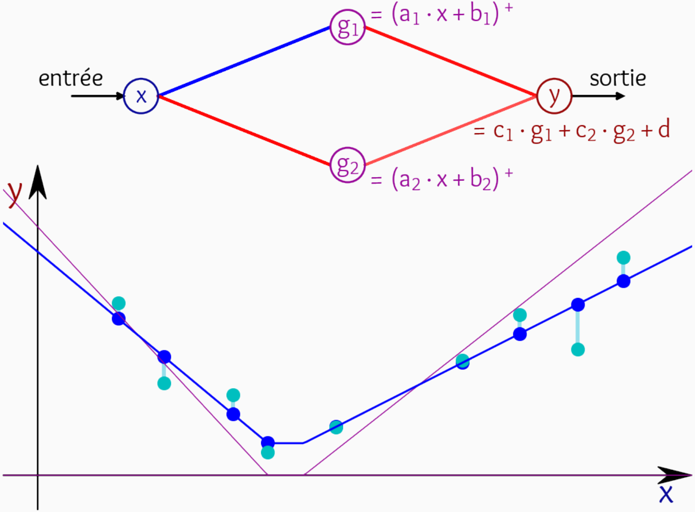
Let's complexify our model with intermediate variables...



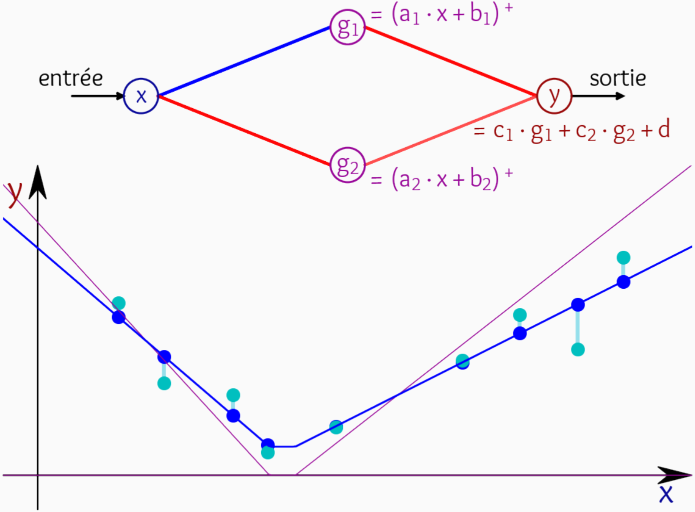
Let’s complexify our model with intermediate variables...



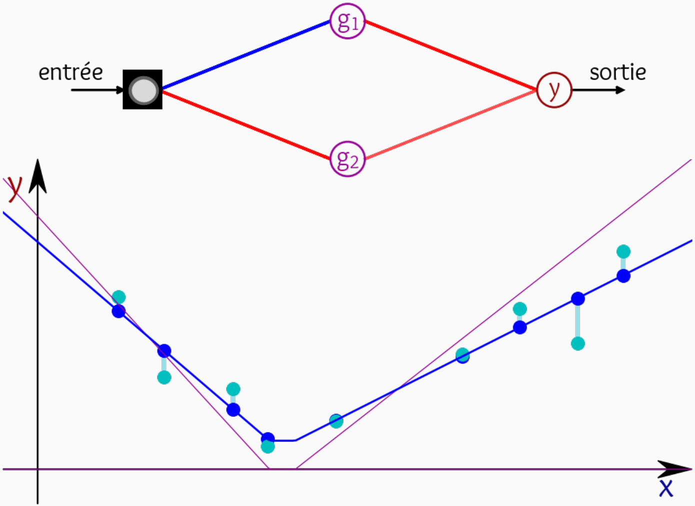
Let's complexify our model with intermediate variables...



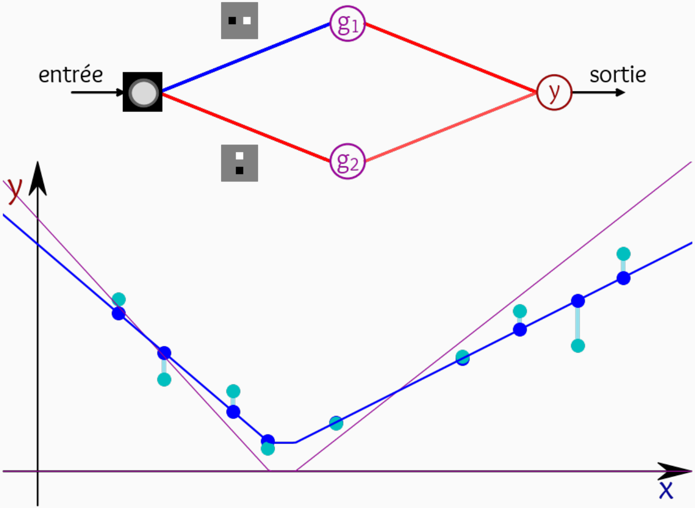
Let's complexify our model with intermediate variables...



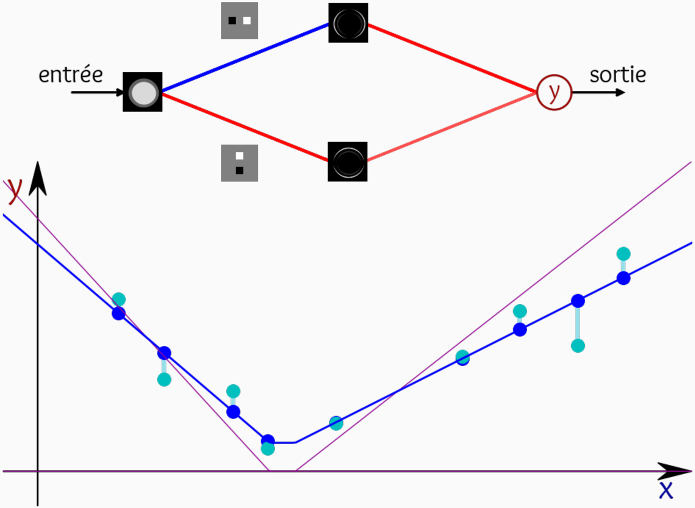
Let's complexify our model with intermediate variables...



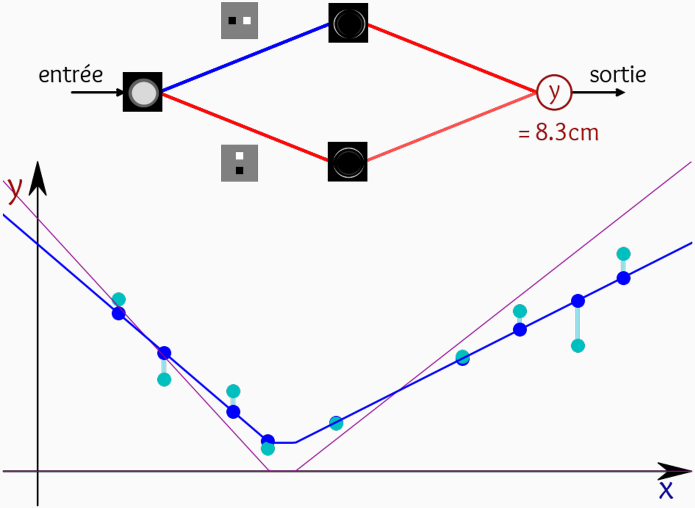
Let's complexify our model with intermediate variables...



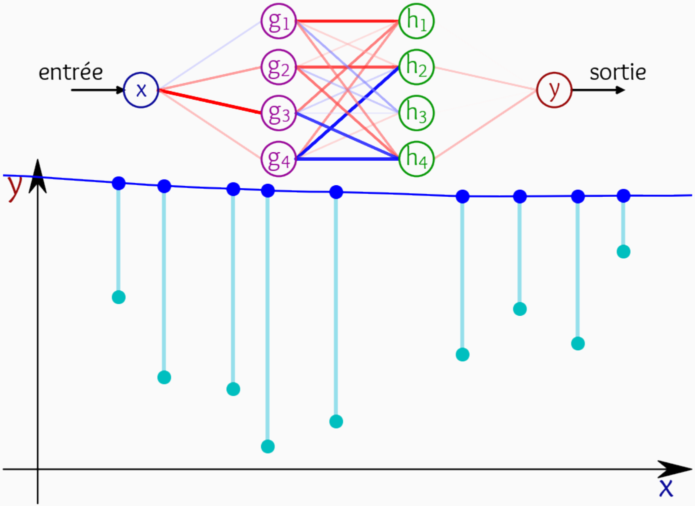
Let's complexify our model with intermediate variables...



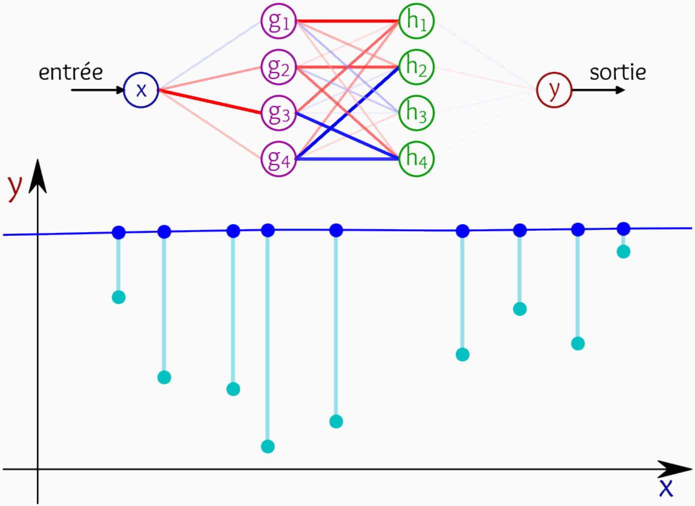
Let's complexify our model with intermediate variables...



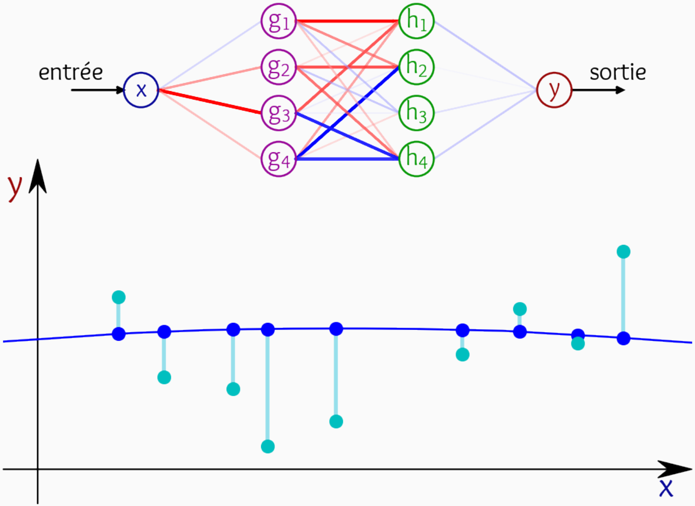
We can then increase the number of “neurons” and layers!



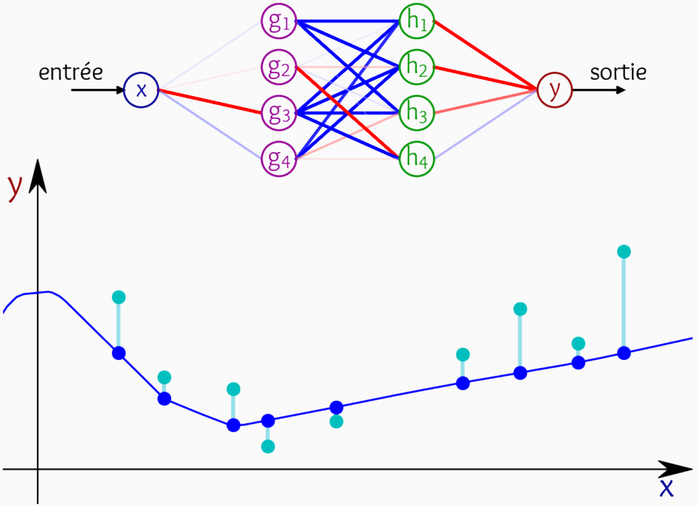
We can then increase the number of “neurons” and layers!



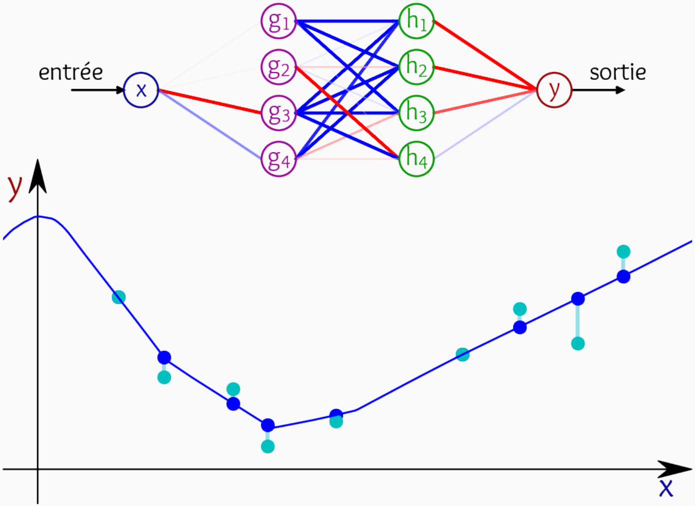
We can then increase the number of “neurons” and layers!



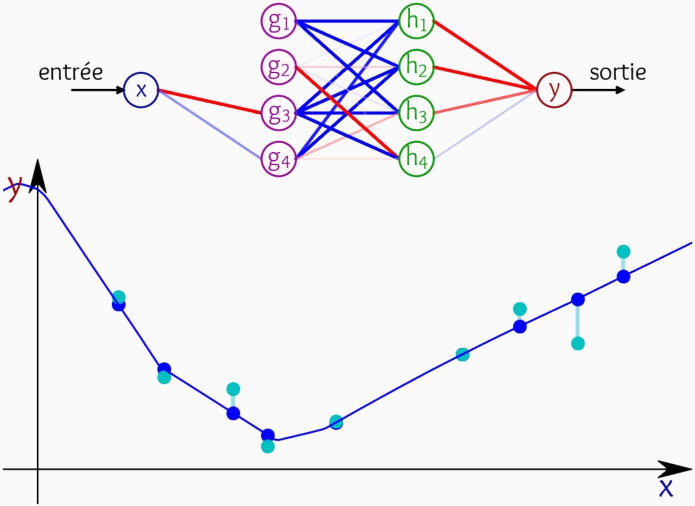
We can then increase the number of “neurons” and layers!



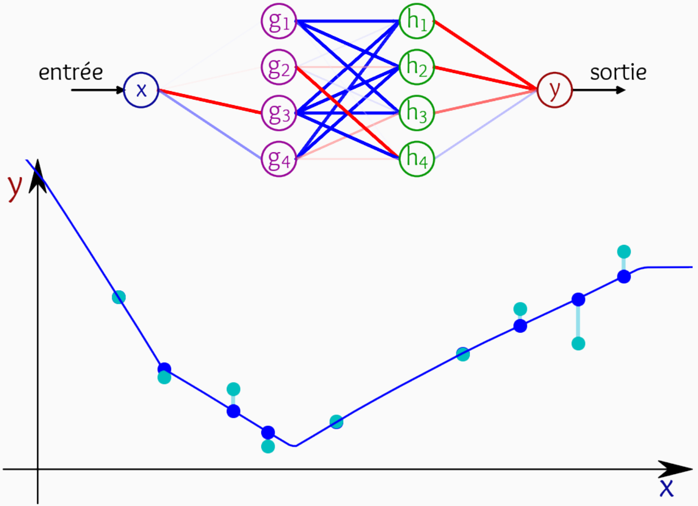
We can then increase the number of “neurons” and layers!



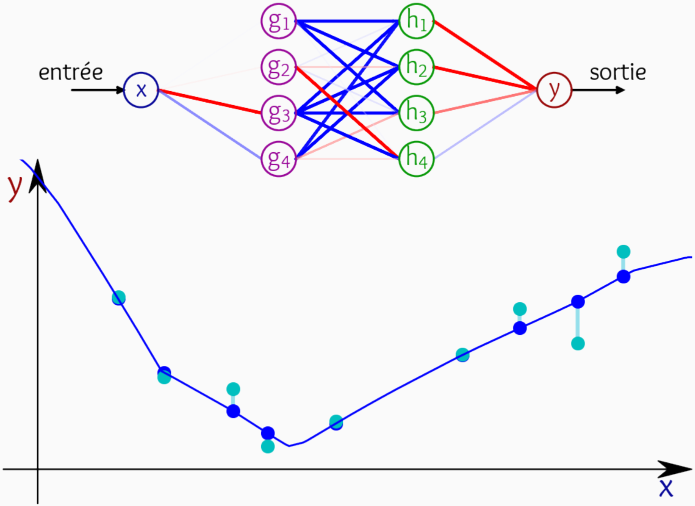
We can then increase the number of “neurons” and layers!



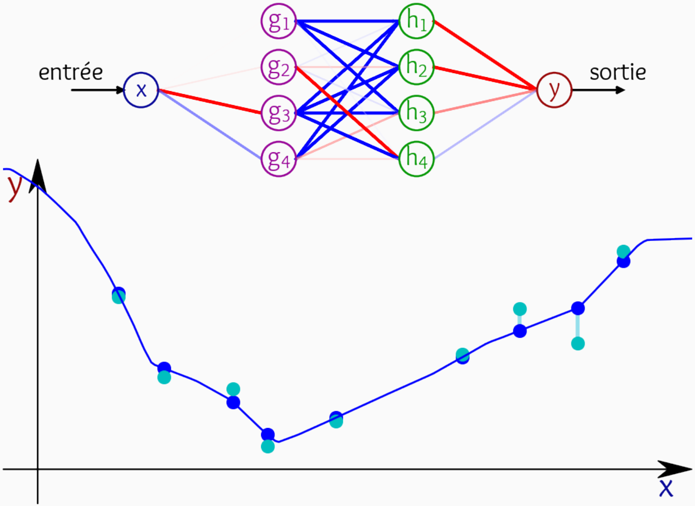
We can then increase the number of “neurons” and layers!



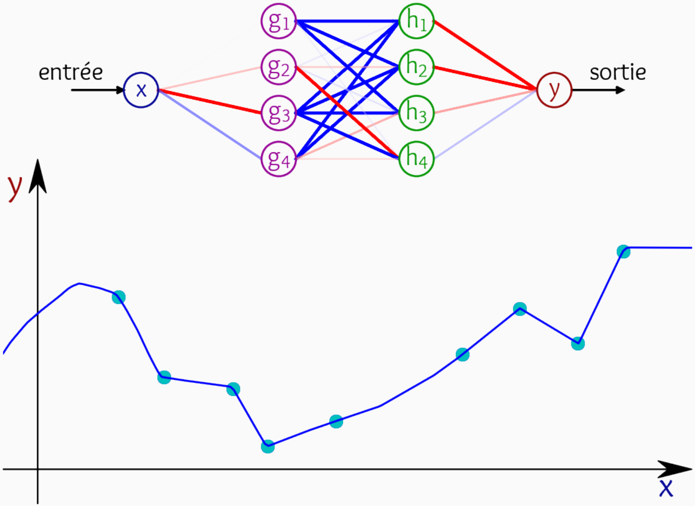
We can then increase the number of "neurons" and layers!



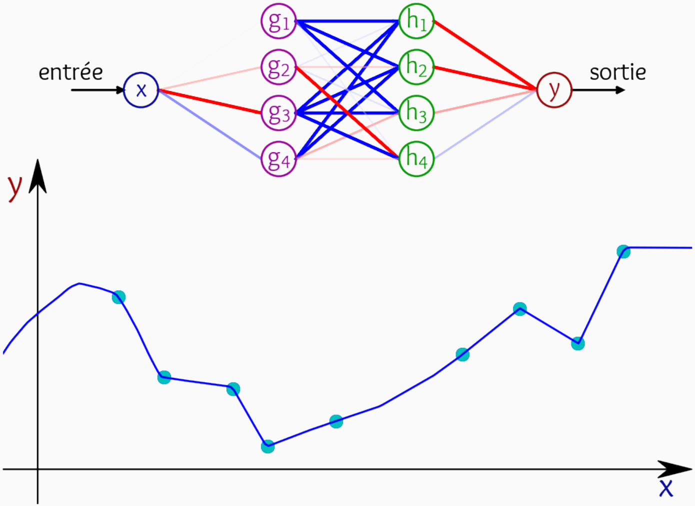
We can then increase the number of "neurons" and layers!



We can then increase the number of "neurons" and layers!



We can then increase the number of “neurons” and layers!



Neural networks:

- Generalization of the **linear regression** to arbitrary models.

Neural networks:

- Generalization of the **linear regression** to arbitrary models.
- Introduces **intermediate variables** and **bendings**.

Neural networks:

- Generalization of the **linear regression** to arbitrary models.
- Introduces **intermediate variables** and **bendings**.
- **Naive** training procedure (flexible rod + springs).

Neural networks:

- Generalization of the **linear regression** to arbitrary models.
- Introduces **intermediate variables** and **bendings**.
- **Naive** training procedure (flexible rod + springs).

Neural networks:

- Generalization of the **linear regression** to arbitrary models.
- Introduces **intermediate variables** and **bendings**.
- **Naive** training procedure (flexible rod + springs).

Generic neural network

$\simeq$ **Interpolation** between the samples of the database.

Neural networks:

- Generalization of the **linear regression** to arbitrary models.
- Introduces **intermediate variables** and **bendings**.
- **Naive** training procedure (flexible rod + springs).

Generic neural network

$\simeq$ **Interpolation** between the samples of the database.

Is it good enough?

Medical imaging $\neq$ generic Big Data problem

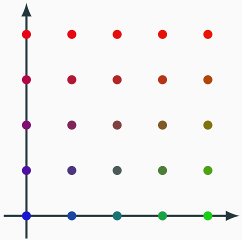
Medical imaging $\neq$ generic Big Data problem

1 number

$\rightarrow$ 5 samples

Medical imaging $\neq$ generic Big Data problem

1 number
$\rightarrow$ 5 samples

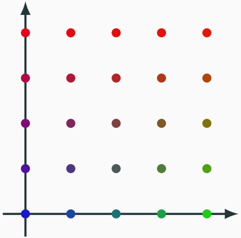


2 numbers
$\rightarrow 5^2$ samples

Medical imaging $\neq$ generic Big Data problem

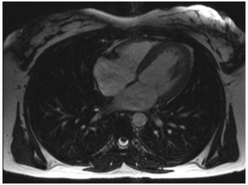
1 number

$\rightarrow$ 5 samples



2 numbers

$\rightarrow 5^2$ samples

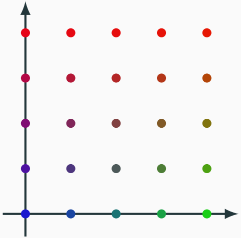


$128 \cdot 128$ numbers

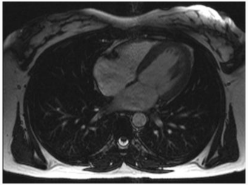
$\rightarrow 5^{128 \cdot 128}$ samples

Medical imaging $\neq$ generic Big Data problem

1 number
$\rightarrow$ 5 samples



2 numbers
$\rightarrow 5^2$ samples



$128 \cdot 128$ numbers
$\rightarrow 5^{128 \cdot 128}$ samples

The set of all 2D/3D images is **way too large**
to be sampled with a satisfying accuracy.

In medical imaging, a good model $F(w; x) \simeq y$ should:

Encode **a sensible prior** on the data

Can I understand a heart MRI as a deformed template?

In medical imaging, a good model $F(w; x) \simeq y$ should:

Encode **a sensible prior** on the data

Can I understand a heart MRI as a deformed template?

Put **constraints** on the decision function,

Thus allowing us to extrapolate **outside** of the training database.

In medical imaging, a good model $F(w; x) \simeq y$ should:

Encode **a sensible prior** on the data

Can I understand a heart MRI as a deformed template?

Put **constraints** on the decision function,

Thus allowing us to extrapolate **outside** of the training database.

Rely on **cheap, elementary operations**,

To scale up to 3D/4D volumes.

In medical imaging, a good model $F(w; x) \simeq y$ should:

Encode **a sensible prior** on the data

Can I understand a heart MRI as a deformed template?

Put **constraints** on the decision function,

Thus allowing us to extrapolate **outside** of the training database.

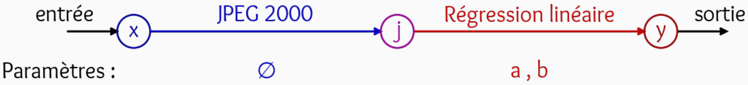
Rely on **cheap, elementary operations**,

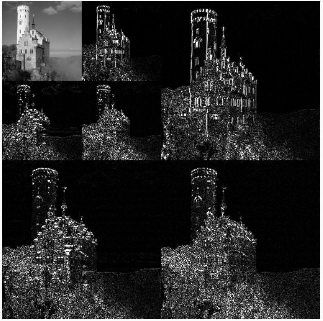
To scale up to 3D/4D volumes.

$\Longrightarrow$ **Let's combine "regression" with "JPEG2000" !**

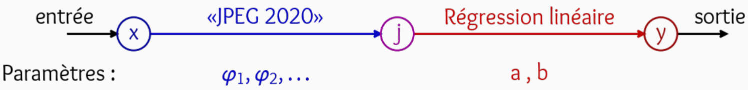
Analysis through multi-scale filtering :
$\Longrightarrow$ Convolutional Neural Networks

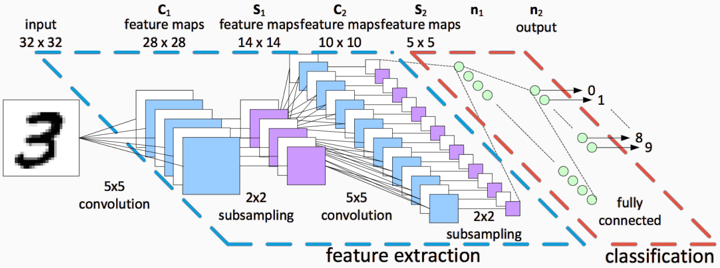
Classical signal processing [Dam] :





Modern signal processing [PMC11] :





Convolutional Neural Networks: an excellent compromise

JPEG2000 relies on a model $F(\mathbf{w}\,;x\,) \simeq y$ that is:

- Computationally cheap.

Convolutional Neural Networks: an excellent compromise

JPEG2000 relies on a model $F(\mathbf{w}\,;x\,) \simeq y$ that is:

- Computationally cheap.
- Constraining

Convolutional Neural Networks: an excellent compromise

JPEG2000 relies on a model $F(\mathbf{w}\,; x\,) \simeq y$ that is:

- Computationally cheap.
- Constraining
- Encodes a **multi-scale** prior on natural images.

Convolutional Neural Networks: an excellent compromise

JPEG2000 relies on a model $F(\mathbf{w}\,;x) \simeq y$ that is:

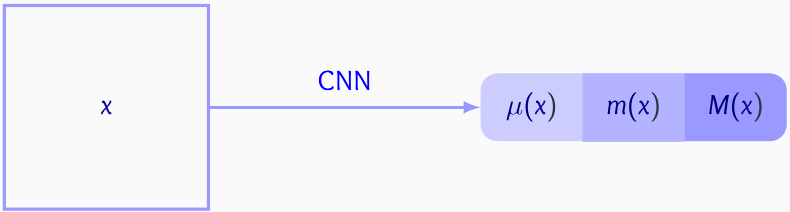
- Computationally cheap.
- Constraining
- Encodes a **multi-scale** prior on natural images.

Convolutional Neural Networks: an excellent compromise

JPEG2000 relies on a model $F(\mathbf{w}\,; x) \simeq y$ that is:

- Computationally cheap.
- Constraining
- Encodes a **multi-scale** prior on natural images.

By **tuning its parameters** on a labeled database,
we get a **CNN** $\simeq$ problem-dependent "JPEG2020".

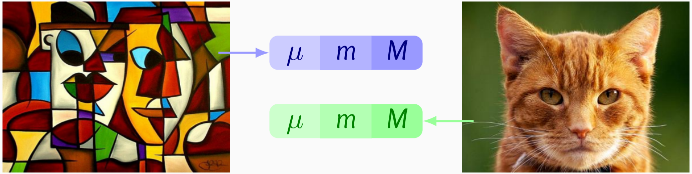


An iconic application: Deep Art [NN16]

An iconic application: Deep Art [NN16]

An iconic application: Deep Art [NN16]

An iconic application: Deep Art [NN16]

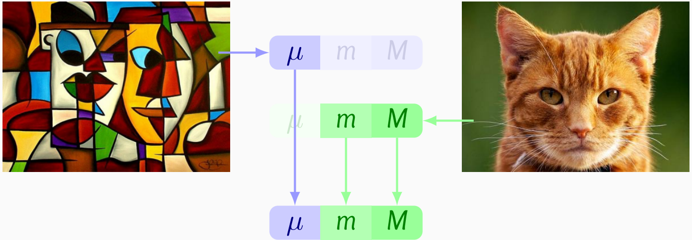


An iconic application: Deep Art [NN16]

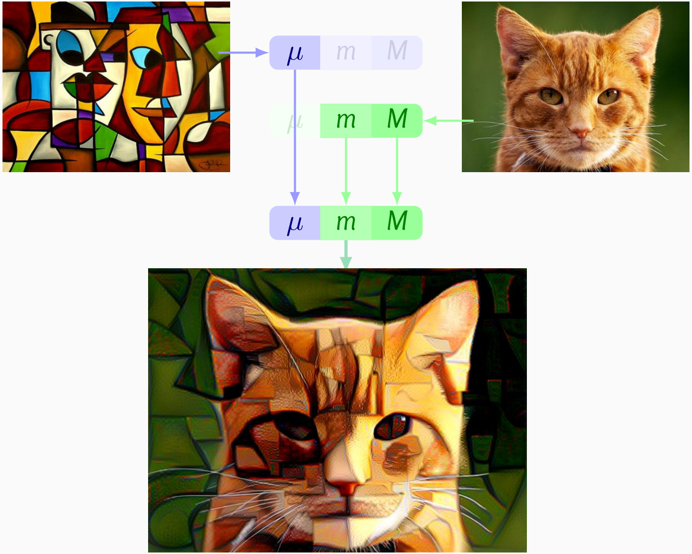
$\mu \quad m \quad M$

$\mu \quad m \quad M$

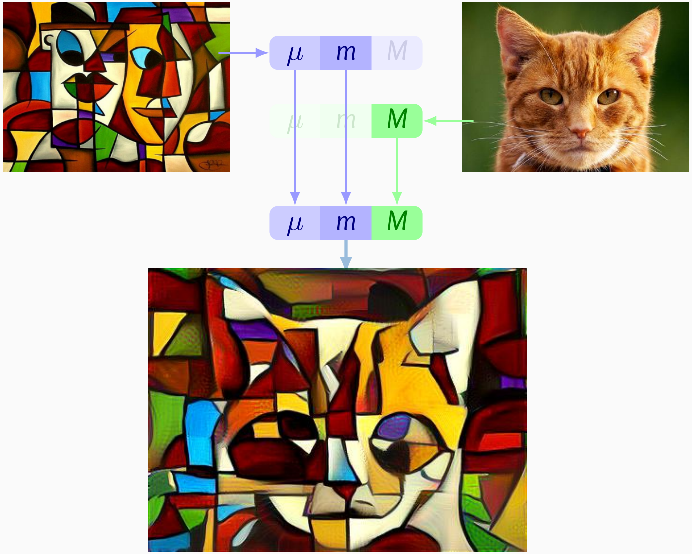
An iconic application: Deep Art [NN16]



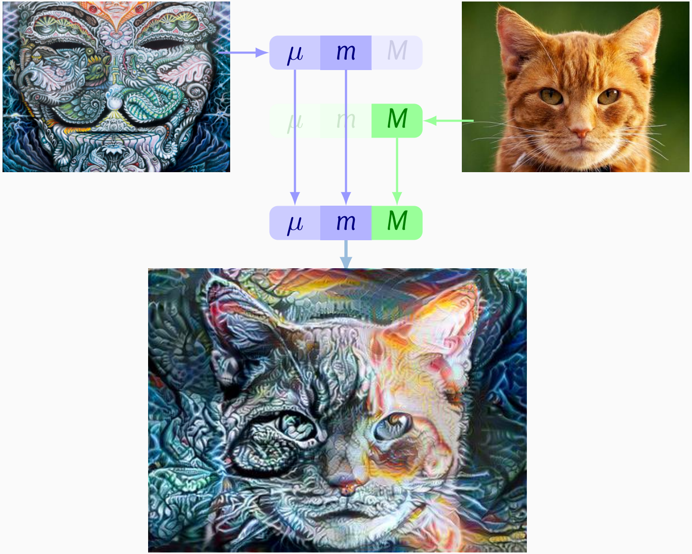
An iconic application: Deep Art [NN16]



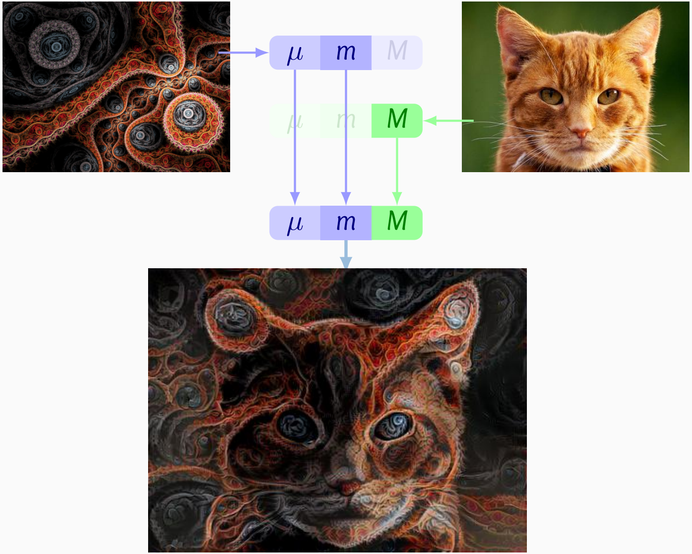
An iconic application: Deep Art [NN16]



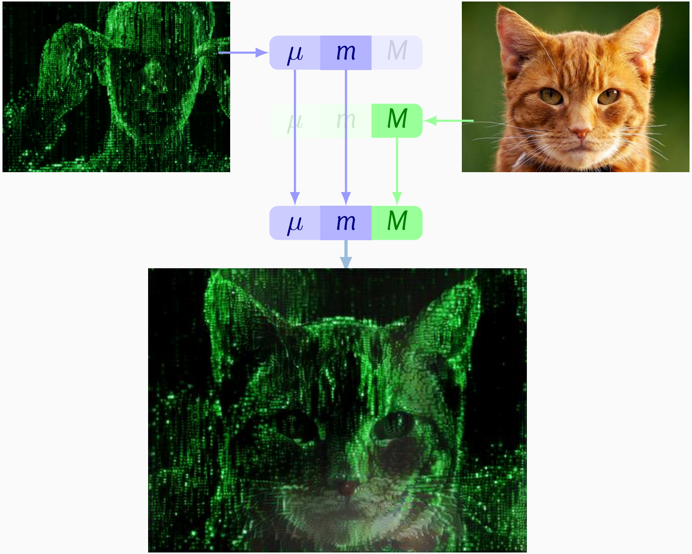
An iconic application: Deep Art [NN16]



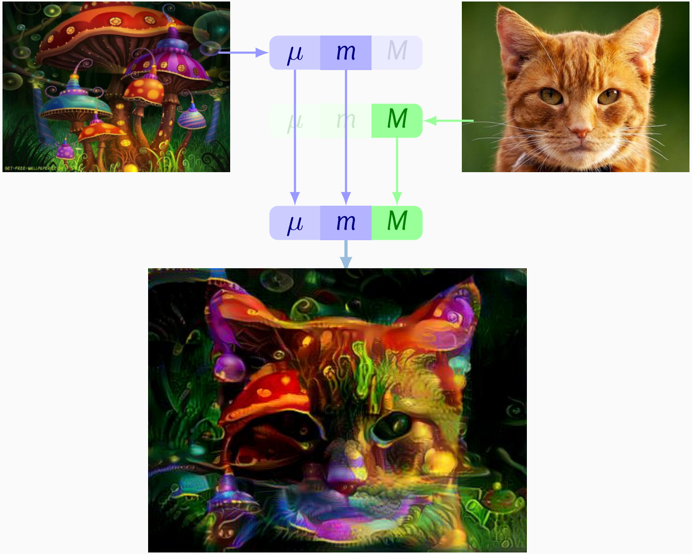
An iconic application: Deep Art [NN16]



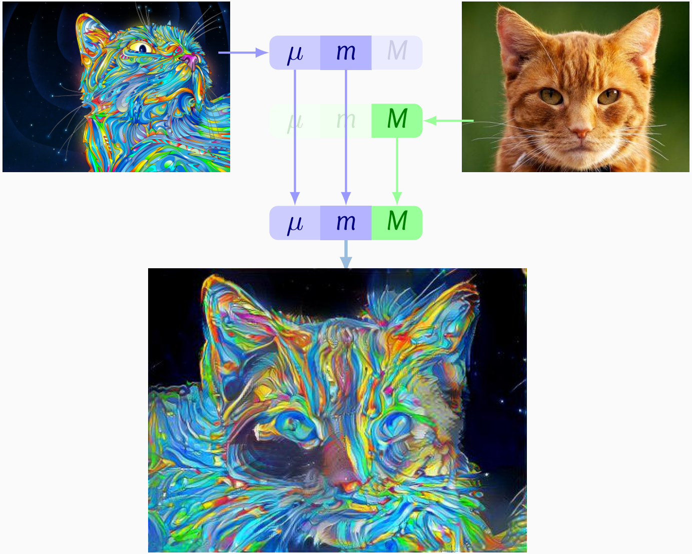
An iconic application: Deep Art [NN16]



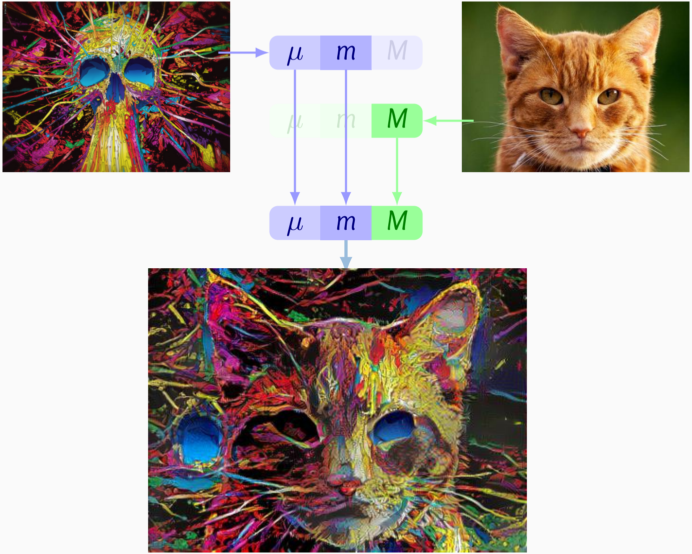
An iconic application: Deep Art [NN16]



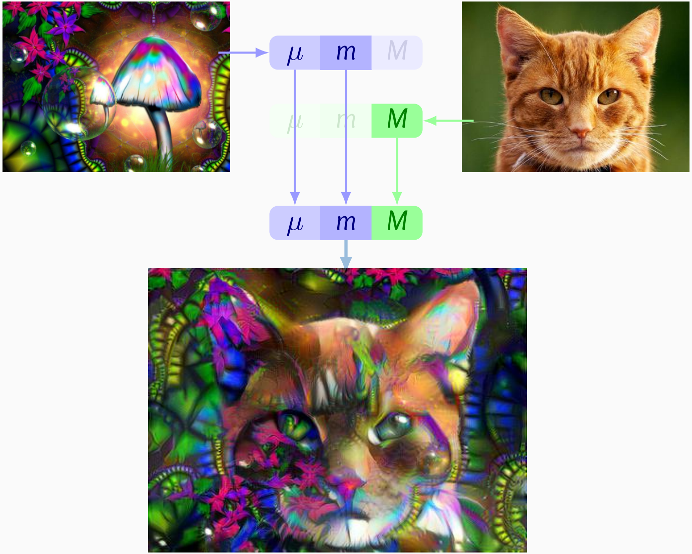
An iconic application: Deep Art [NN16]



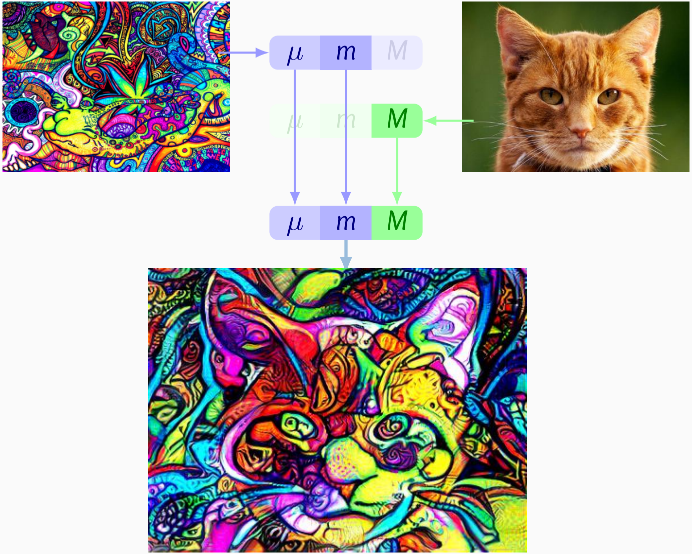
An iconic application: Deep Art [NN16]



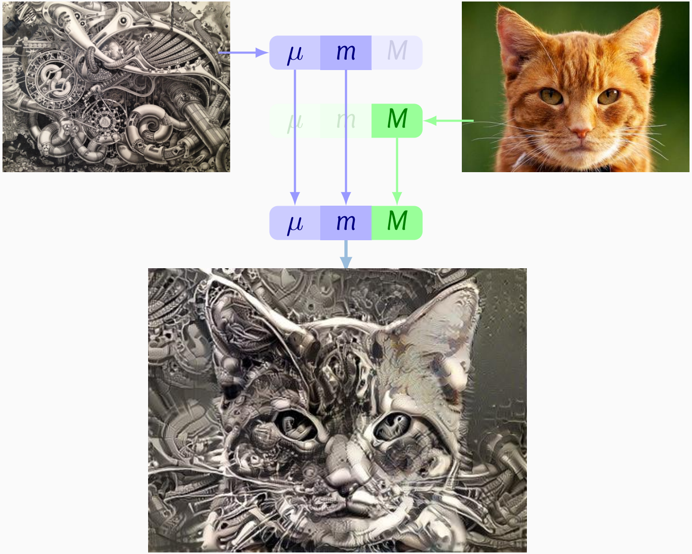
An iconic application: Deep Art [NN16]



An iconic application: Deep Art [NN16]



An iconic application: Deep Art [NN16]



The dreamed application: image classification

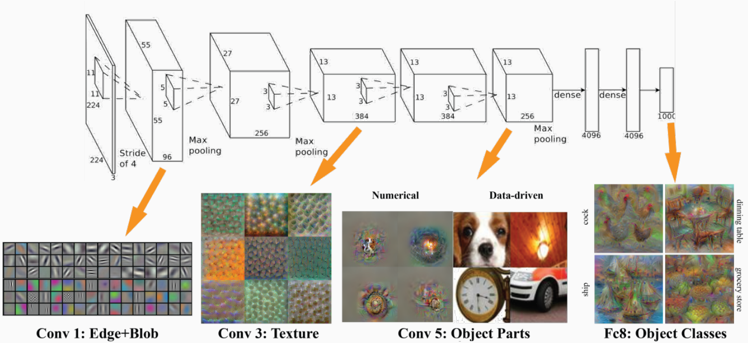
Looking at $\mathrm{CNN}(x) = [\,\mu(x)\,,\ m(x)\,,\ M(x)\,]$,

can we **separate** seagulls from pandas?

The dreamed application: image classification

Looking at $\mathrm{CNN}(x) = [\,\mu(x)\,,\ m(x)\,,\ M(x)\,]$,
can we **separate** seagulls from pandas?

What researchers have in mind [WZTF]:



$\mu(x)$ $\quad m(x)$ $\quad M(x)$

The limits of multi-scale filtering

CNNs perform **feature detection**, nothing more, nothing less [NYC15]:

The limits of multi-scale filtering

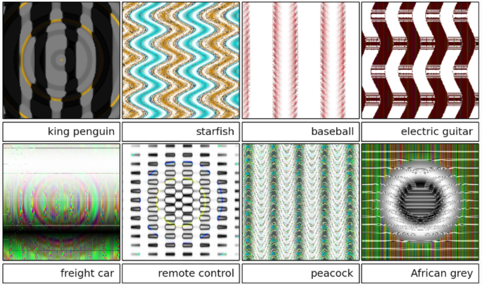
CNNs perform **feature detection**, nothing more, nothing less [NYC15]:

« $\mu(x)$ is reliable ; $M(x)$ really isn't. »

The limits of multi-scale filtering

CNNs perform **feature detection**, nothing more, nothing less [NYC15]:

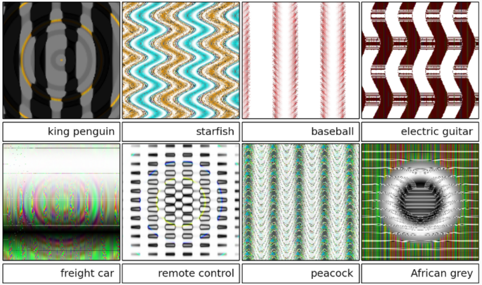
« $\mu(x)$ is reliable ; $M(x)$ really isn't. »



The limits of multi-scale filtering

CNNs perform **feature detection**, nothing more, nothing less [NYC15]:

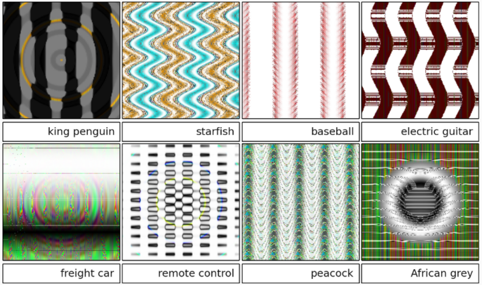
« $\mu(x)$ is reliable ; $M(x)$ really isn't. »



The limits of multi-scale filtering

CNNs perform **feature detection**, nothing more, nothing less [NYC15]:

« $\mu(x)$ is reliable ; $M(x)$ really isn't. »



Unfortunately : **structured** anatomical models are **way** more expensive.

(that's my job...)

Conclusion

Old ideas, fancy words

As we've seen :

Multi-layer perceptron $\Longleftrightarrow$ Neural network

Old ideas, fancy words

As we've seen :

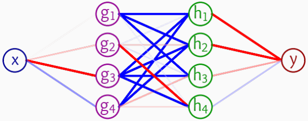
Multi-layer perceptron	$\Longleftrightarrow$	Neural network
Regression on a multi-resolution bank of filters	$\Longleftrightarrow$	Deep learning

Old ideas, fancy words

As we've seen :

Multi-layer perceptron $\Longleftrightarrow$ Neural network

Regression on a multi-resolution bank of filters $\Longleftrightarrow$ Deep learning



$\overset{?}{\Longleftrightarrow}$

Old ideas, fancy words

As we've seen :

Multi-layer perceptron	$\Longleftrightarrow$	Neural network
Regression on a multi-resolution bank of filters	$\Longleftrightarrow$	Deep learning

$\overset{?}{\Longleftrightarrow}$

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~
- The first convincing model for **texture**.

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~
- The first convincing model for **texture**.
- The development of **high-level software tools** that allow us to **tune the parameters** of our models – TensorFlow, PyTorch...

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~
- The first convincing model for **texture**.
- The development of **high-level software tools** that allow us to **tune the parameters** of our models – TensorFlow, PyTorch...

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~
- The first convincing model for **texture**.
- The development of **high-level software tools** that allow us to **tune the parameters** of our models – TensorFlow, PyTorch...

An image processing software **always** relies on a **simplistic model** of the data.

Conclusion

The Deep Learning revolution is all about:

- ~~Artificial intelligence.~~
- The first convincing model for **texture**.
- The development of **high-level software tools** that allow us to **tune the parameters** of our models – TensorFlow, PyTorch...

An image processing software **always** relies on a **simplistic model** of the data.

There is no miracle.

Keep in mind

When facing an “AI” product :

- **« Show me what doesn’t work. »**

Keep in mind

When facing an "AI" product :

- **« Show me what doesn't work. »**
- « Can you **explain** this error to me? »

Keep in mind

When facing an "AI" product :

- **« Show me what doesn't work. »**
- « Can you **explain** this error to me? »
- "More data" **is not** going to solve problems automagically.

Keep in mind

When facing an "AI" product :

- **« Show me what doesn't work. »**
- « Can you **explain** this error to me? »
- "More data" **is not** going to solve problems automagically.

Keep in mind

When facing an "AI" product :

- **« Show me what doesn't work. »**
- « Can you **explain** this error to me? »
- "More data" **is not** going to solve problems automagically.

Going further :

Cours de **Sciences des données** au Collège de France,

`www.college-de-france.fr/site/stephane-mallat/`

Jetez un œil à la leçon inaugurale, très accessible !

AI+Radiology MasterClass

sites.google.com/view/masterclassiaimagerie/home

My notebooks are available: www.math.ens.fr/~feydy/Teaching/

Thank you for your attention.

Any questions ?

References i

Allessio Damato.
Jpeg 2000 — wikipedia, the free encyclopedia.
http://en.wikipedia.org/wiki/JPEG_2000.
Accessed: 2018-05-10.

Olivier Ecabert, Jochen Peters, and Matthew Walker.
Segmentation of the heart and great vessels in ct images using a model-based adaptation framework.
Medical Image Analysis, (15):863–876, 2011.

Stéphane Mallat.
Understanding deep convolutional networks.
Phil. Trans. R. Soc. A, 374(2065):20150203, 2016.

Tomaso Mansi.
A statistical model for quantification and prediction of cardiac remodelling: Application to tetralogy of fallot.
IEEE transactions on medical imaging, 2011.

Yaroslav Nikulin and Roman Novak.
Exploring the neural algorithm of artistic style.
arXiv preprint arXiv:1602.07188, 2016.

Anh Nguyen, Jason Yosinski, and Jeff Clune.
Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

Maurice Peemen, Bart Mesman, and Henk Corporaal.
Speed sign detection and recognition by convolutional neural networks.
In *Proceedings of the 8th International Automotive Congress*, pages 162–170, 2011.

References iv

Donglai Wei, Bolei Zhou, Antonio Torralba, and William Freeman.
mneuron: A matlab plugin to visualize neurons from deep models.
http://vision03.csail.mit.edu/cnn_art/.
Accessed: 2018-05-10.