

“Réseaux de neurones” artificiels

Les questions à poser

Jean Feydy

DIU de Neuroradiologie Vieillesse – 15 mai 2020

Imperial College London

Jean Feydy :

- Thèse avec Alain Trouvé, ENS Cachan + PostDoc à Londres,
Recalage d'images médicales.
- TDs pour Gabriel Peyré, M2 MVA et ENS Ulm,
Fondements mathématiques des Data Sciences.
- Fils d'Antoine Feydy, PUPH à Cochin,
Spécialisé en **imagerie musculo-squelettique.**

Réseaux de neurones?

Réseaux de neurones?

Une généralisation de la **régression linéaire**
à des modèles plus complexes.

Réseaux de neurones?

Une généralisation de la **régression linéaire**
à des modèles plus complexes.

Et ça marche? En général, **non**.

Réseaux de neurones?

Une généralisation de la **régression linéaire**
à des modèles plus complexes.

Et ça marche? En général, **non**.

Par contre, cette idée permet aujourd'hui de produire
d'excellents **détecteurs de motifs**.

Réseaux de neurones?

Une généralisation de la **régression linéaire**
à des modèles plus complexes.

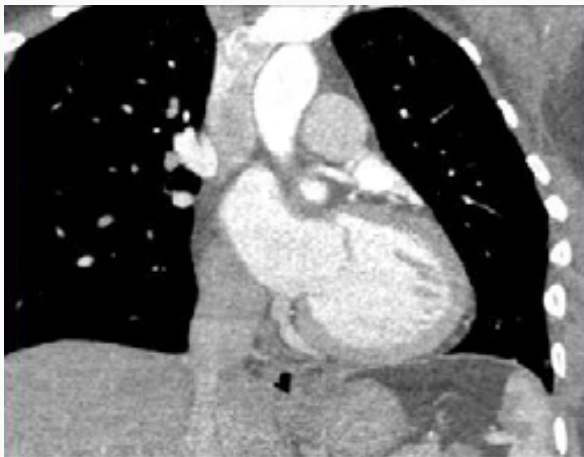
Et ça marche? En général, **non**.

Par contre, cette idée permet aujourd'hui de produire
d'excellents **détecteurs de motifs**.

En tant que médecins, quel **regard critique**
pouvez-vous porter sur ces techniques?

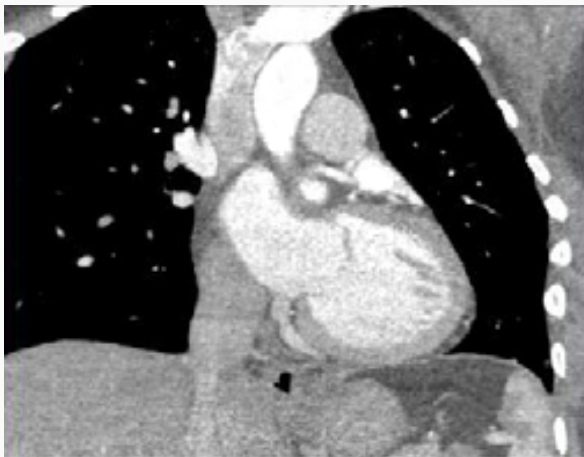
Que lit-on sur une radio?

Une image : trois niveaux d'analyse [EPW11, Man11]



Une image : trois niveaux d'analyse [EPW11, Man11]

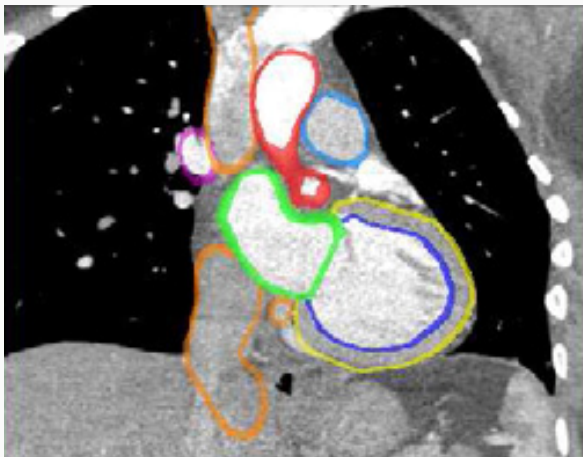
1. Texture



Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

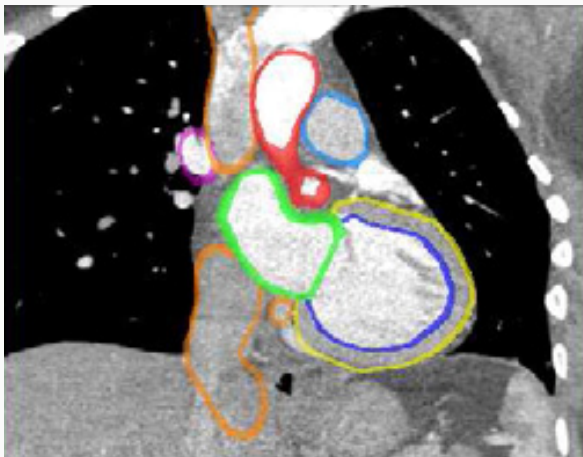
2. Anatomie



Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

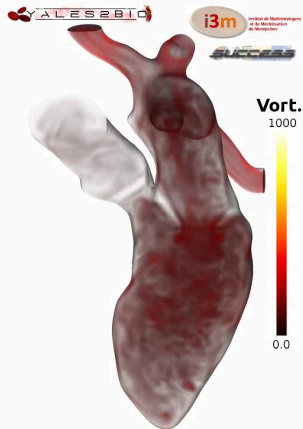


Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



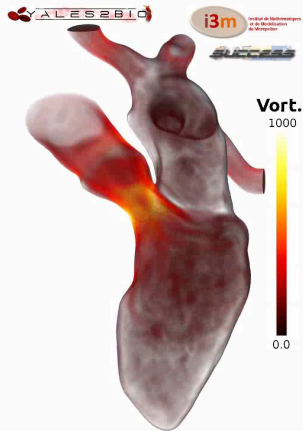
Time: 0 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



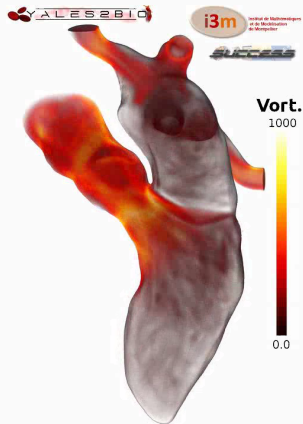
Time: 100 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



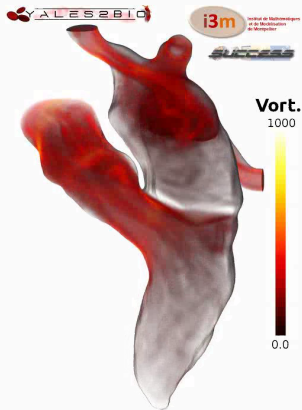
Time: 200 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



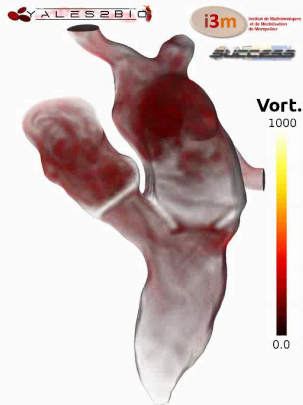
Time: 300 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



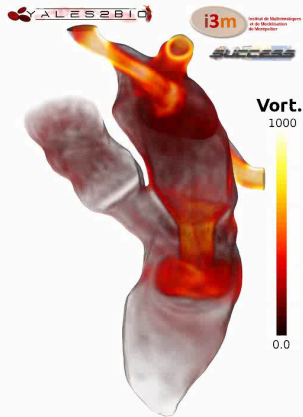
Time: 400 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



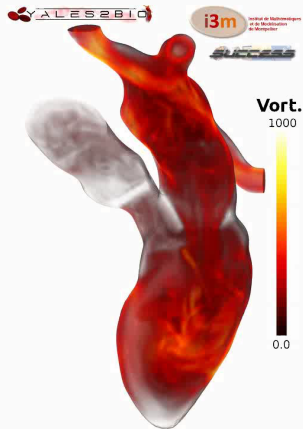
Time: 500 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



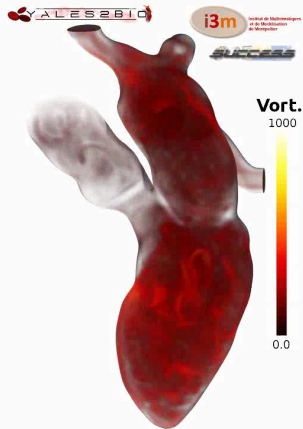
Time: 600 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



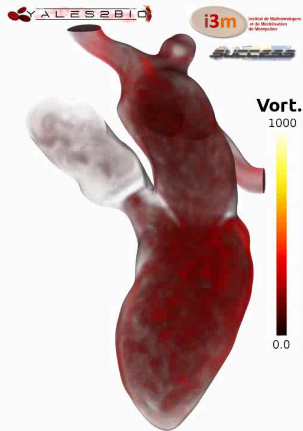
Time: 700 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



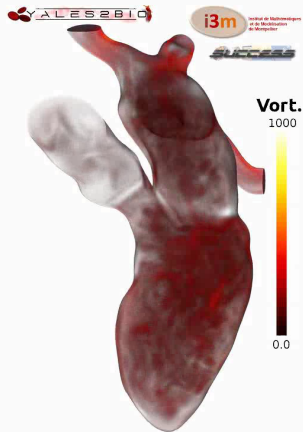
Time: 800 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



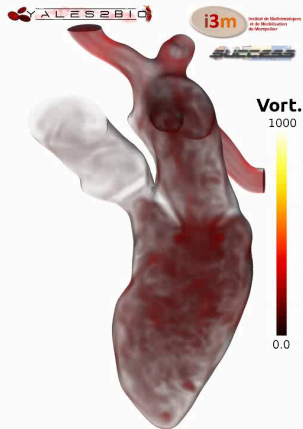
Time: 900 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction



Time: 0 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Texture

2. Anatomie

3. Fonction

Chacun de ces niveaux d'abstraction peut être **modélisé** en se reposant sur le précédent.

1. Texture

2. Anatomie

3. Fonction

Chacun de ces niveaux d'abstraction peut être
modélisé en se reposant sur le précédent.

Découvrons la plus fondamentale de toutes ces théories :
l'analyse par **filtrage multi-résolution**.

Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



Le filtrage, ou “produit de convolution”

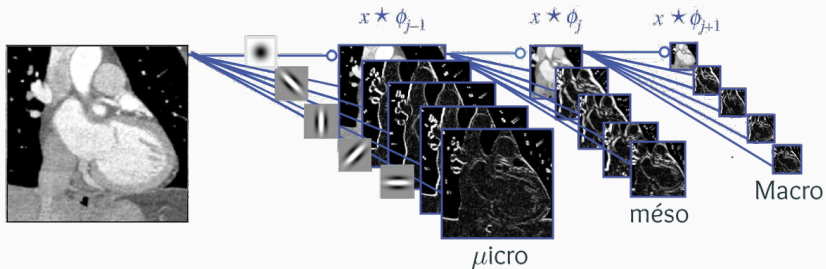
Convolution (i.e. moyenne pondérée des pixels voisins) :
Généralisation peu coûteuse du produit “ $a \cdot x$ ”,
paramétrée par les coefficients d’un petit filtre φ .



A priori multi-échelles sur les images

Théorie des **ondelettes** (Meyer, Mallat, Daubechies...) :

Petits filtres + dé-zooms en cascade [Mal16] :



A priori multi-échelles sur les images

Théorie des **ondelettes** (Meyer, Mallat, Daubechies...) :

Petits filtres + dé-zooms en cascade [Mal16] :

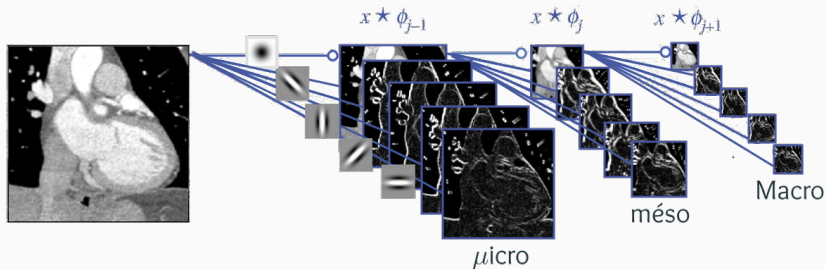


Image \longrightarrow Coefficients pertinents
 \simeq Audio “.wav” \longrightarrow Partition

A priori multi-échelles sur les images

Théorie des **ondelettes** (Meyer, Mallat, Daubechies...) :

Petits filtres + dé-zooms en cascade [Mal16] :

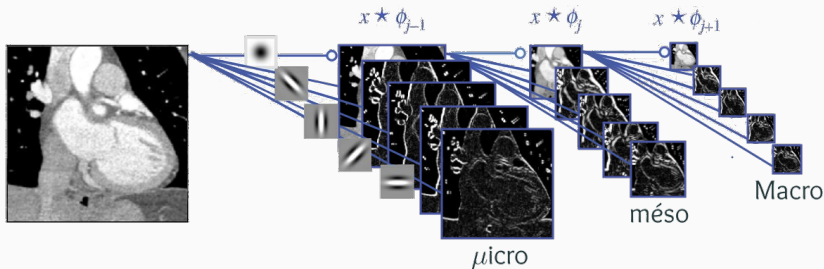


Image \longrightarrow Coefficients pertinents
 \simeq Audio “.wav” \longrightarrow Partition

\implies Format **JPEG 2000**, standard du cinéma numérique.

**Voilà pour les modèles classiques.
Quid des réseaux de neurones?**

“Apprentissage supervisé” = régression

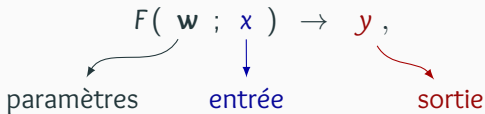
On dispose :

- D'une base de données $\{(x_1 \rightarrow y_1), (x_2 \rightarrow y_2), \dots\}$.

“Apprentissage supervisé” = régression

On dispose :

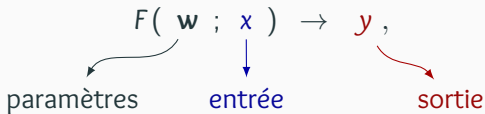
- D'une base de données $\{ (x_1 \rightarrow y_1), (x_2 \rightarrow y_2), \dots \}$.
- D'un modèle explicatif



“Apprentissage supervisé” = régression

On dispose :

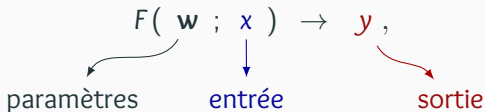
- D'une base de données $\{ (x_1 \rightarrow y_1), (x_2 \rightarrow y_2), \dots \}$.
- D'un modèle explicatif



“Apprentissage supervisé” = régression

On dispose :

- D'une base de données $\{ (x_1 \rightarrow y_1), (x_2 \rightarrow y_2), \dots \}$.
- D'un modèle explicatif

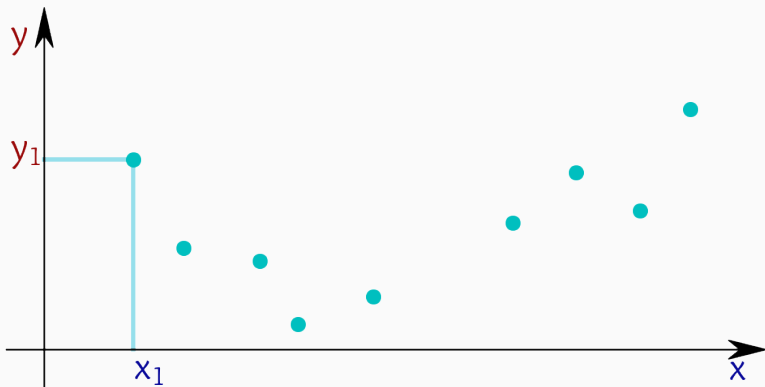


Trouvons, pas-à-pas, un choix $\mathbf{w}_{\text{optimal}}$ des paramètres qui minimise
l'erreur moyenne sur les prédictions.

Un jeu de donnée jouet, en dimension 1

entrée x

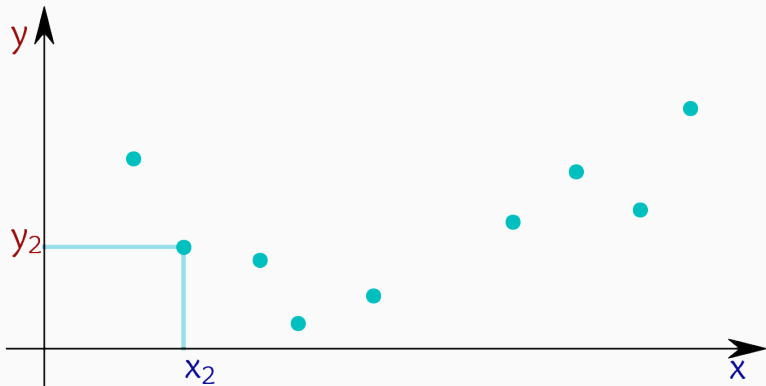
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

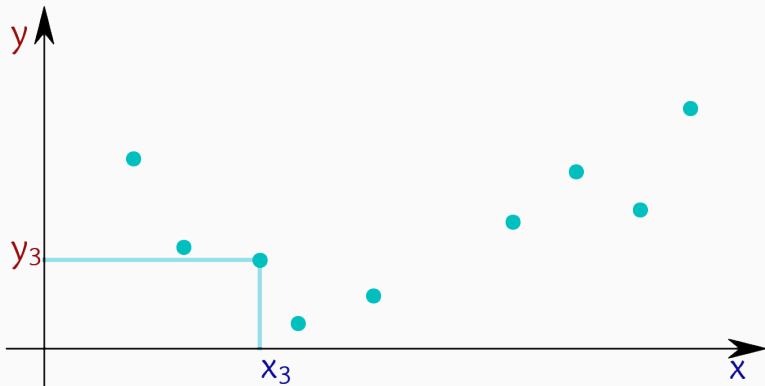
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

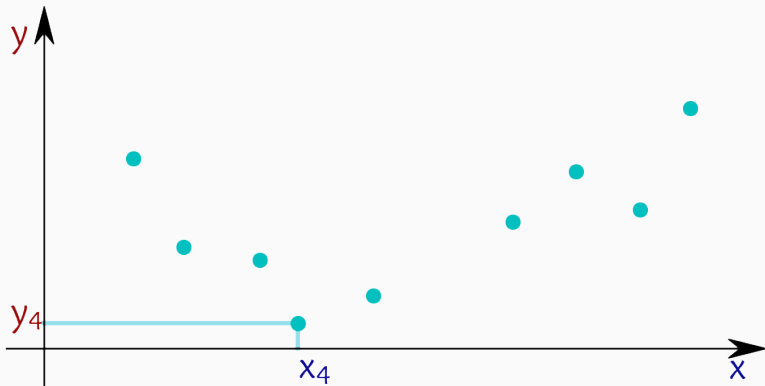
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

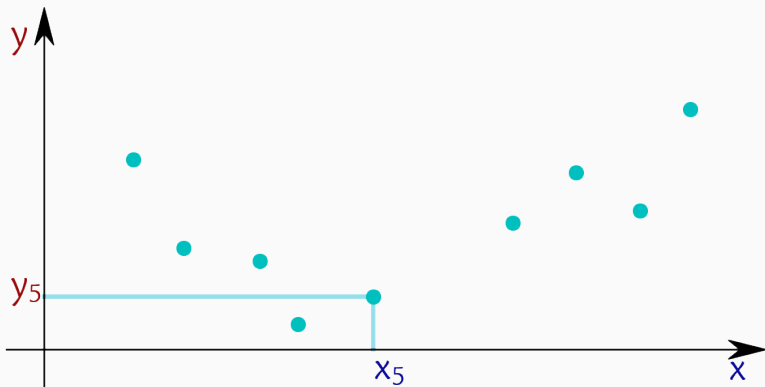
y sortie



Un jeu de donnée jouet, en dimension 1

entrée x

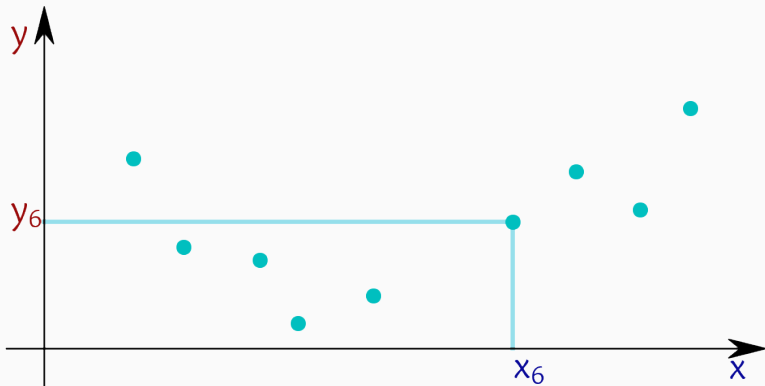
y sortie



Un jeu de donnée jouet, en dimension 1

entrée x

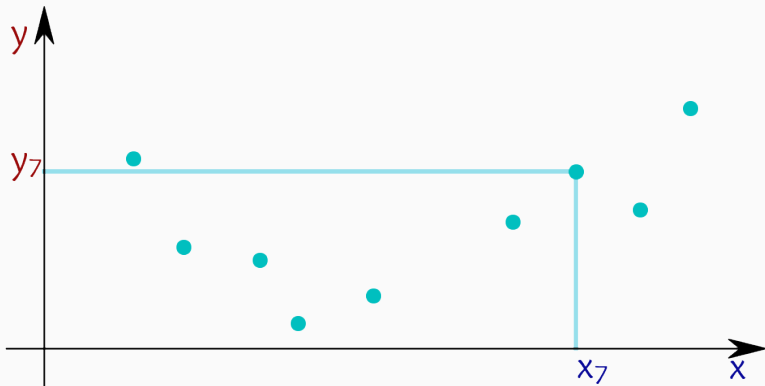
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

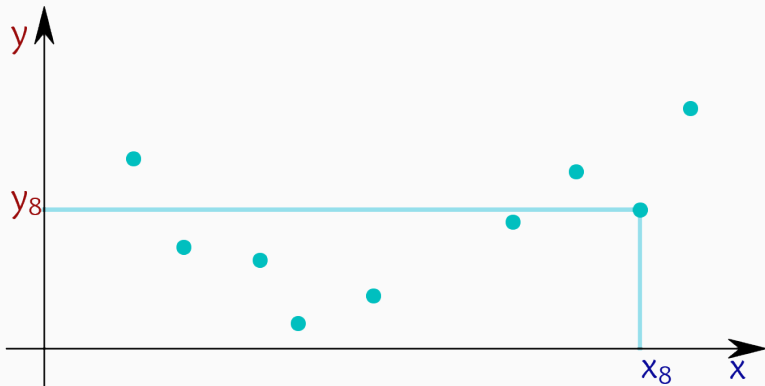
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

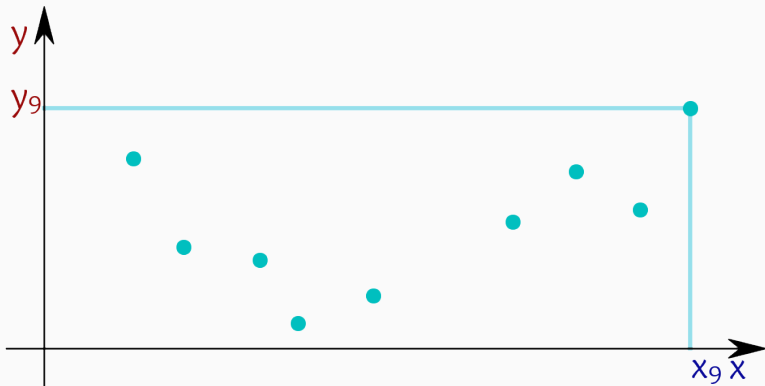
sortie y



Un jeu de donnée jouet, en dimension 1

entrée x

sortie y

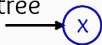


Exemple fondamental : la régression linéaire

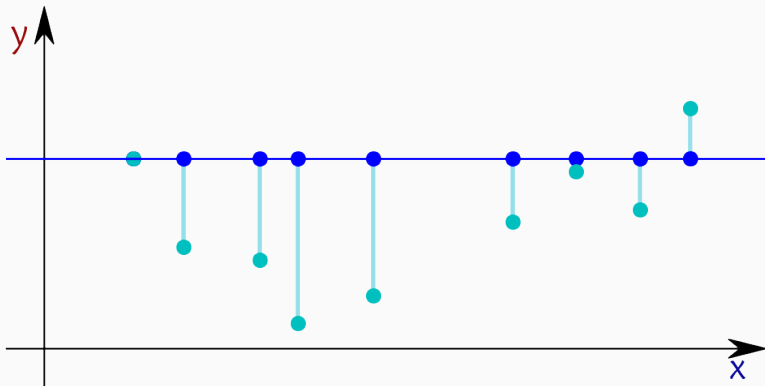
$$F(a, b; x) = a \cdot x + b$$

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

entrée



sortie

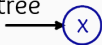


Exemple fondamental : la régression linéaire

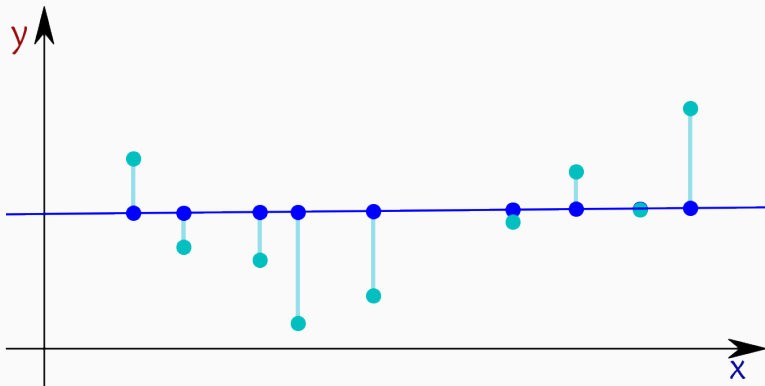
$$F(a, b; x) = a \cdot x + b$$

$$(a, b) = +0.01, +0.18$$

entrée



sortie

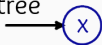


Exemple fondamental : la régression linéaire

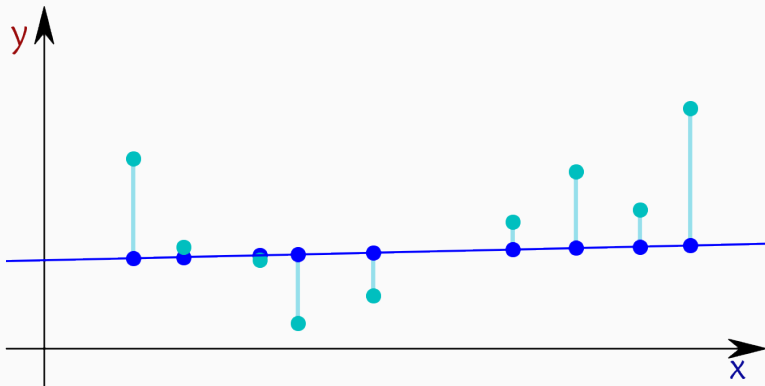
$$F(a, b; x) = a \cdot x + b$$

$$(a, b) = +0.02, +0.12$$

entrée



sortie

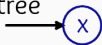


Exemple fondamental : la régression linéaire

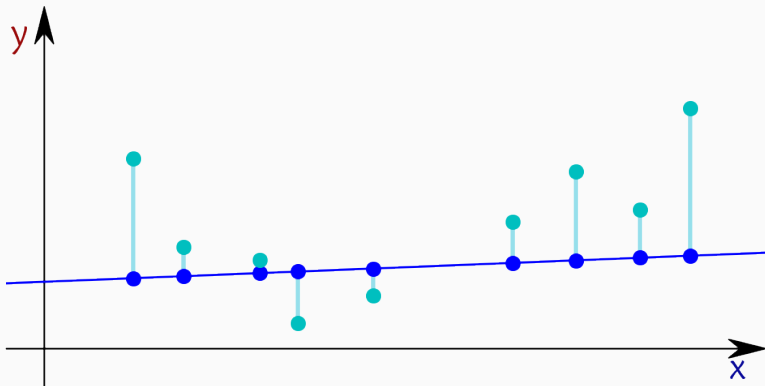
$$F(a, b; x) = a \cdot x + b$$

$$(a, b) = +0.04, +0.09$$

entrée



sortie



Exemple fondamental : la régression linéaire

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

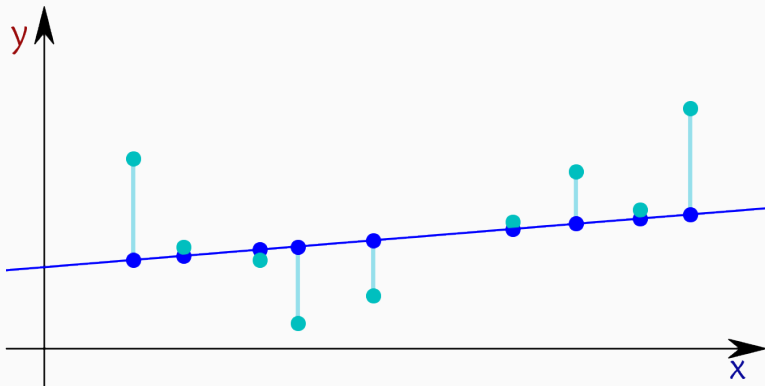
$$(a, b) = +0.08, +0.11$$

entrée

x

sortie

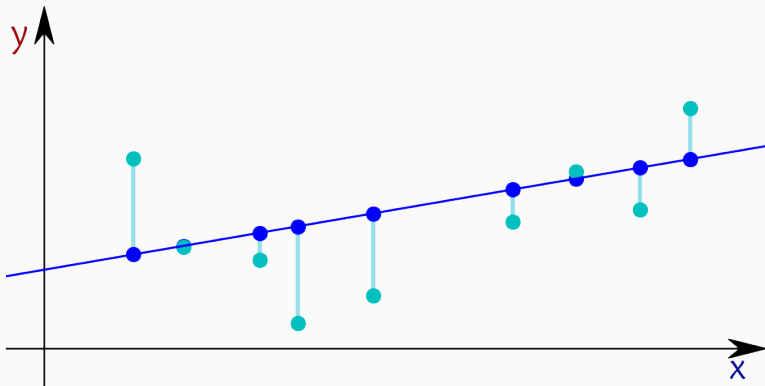
y



Exemple fondamental : la régression linéaire

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

$$(a, b) = +0.17, +0.10$$

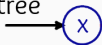


Exemple fondamental : la régression linéaire

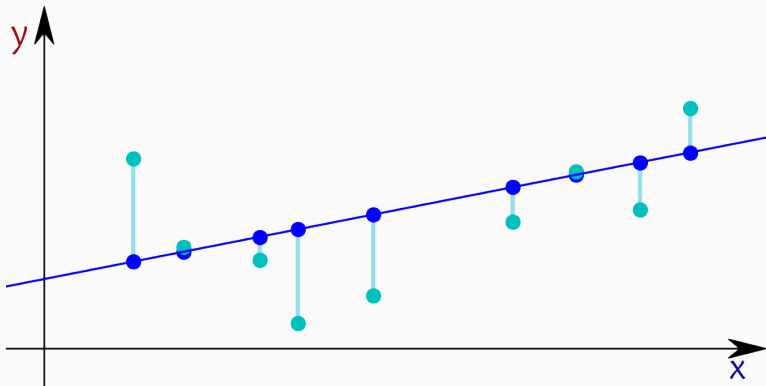
$$F(a, b; x) = a \cdot x + b$$

$$(a, b) = +0.20, +0.09$$

entrée



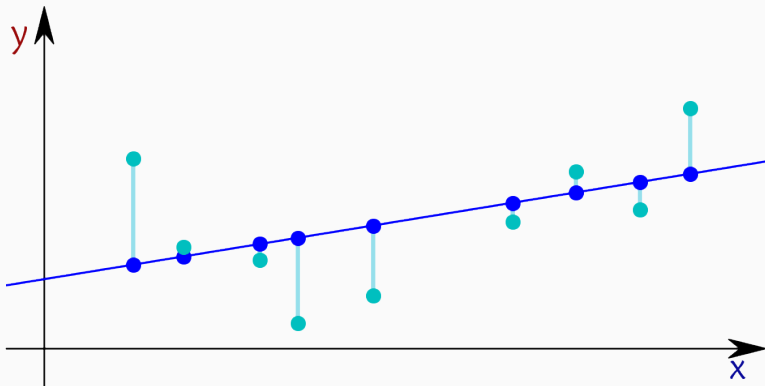
sortie



Exemple fondamental : la régression linéaire

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

$$(a, b) = +0.16, +0.09$$



Exemple fondamental : la régression linéaire

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

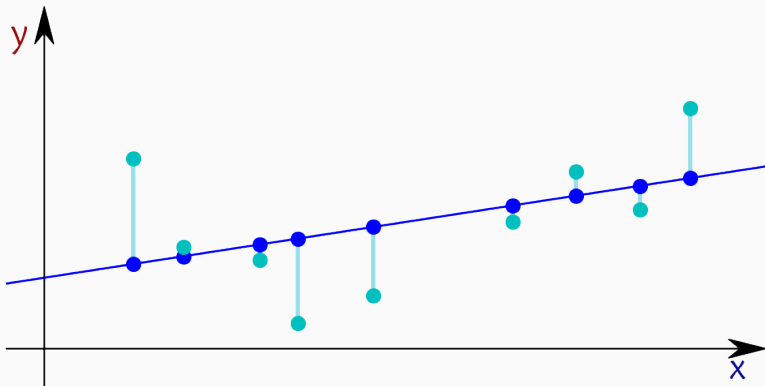
$$(a, b) = +0.15, +0.09$$

entrée

x

sortie

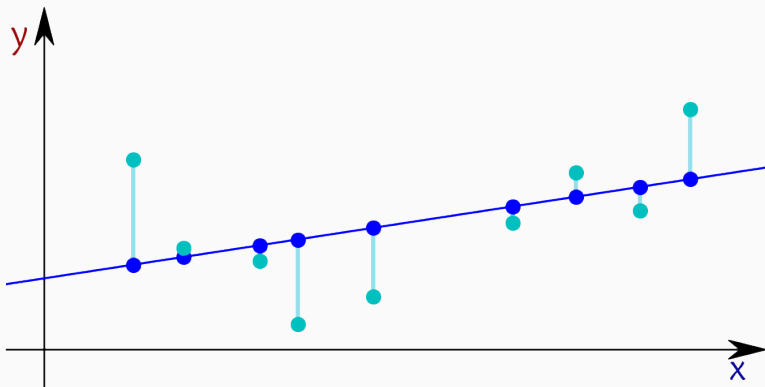
y



Exemple fondamental : la régression linéaire

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

$$(a, b) = +0.15, +0.09$$

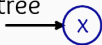


Un modèle plus complexe : la régression quadratique

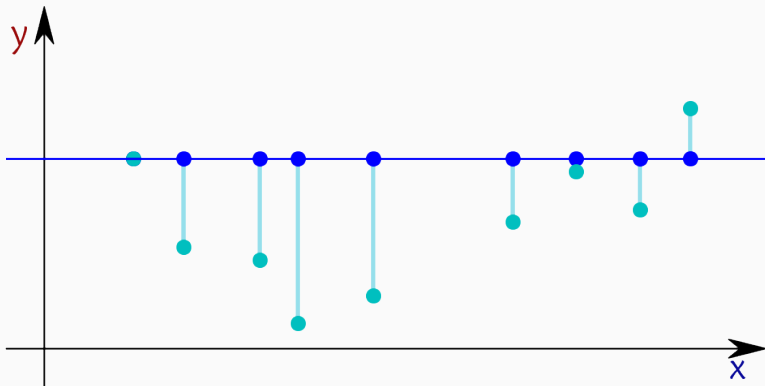
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

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

entrée



sortie

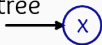


Un modèle plus complexe : la régression quadratique

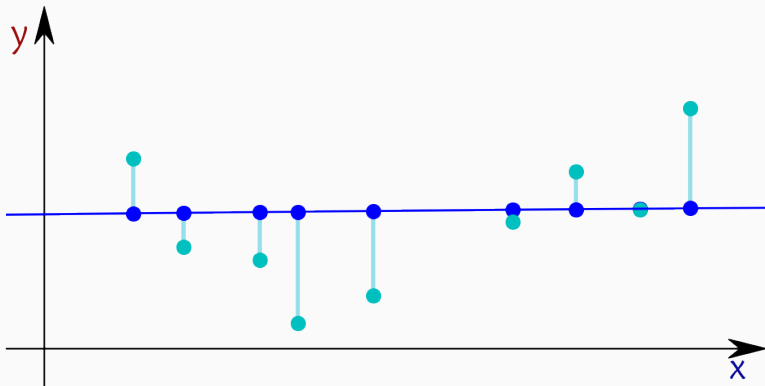
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = -0.00, +0.01, +0.18$$

entrée



sortie

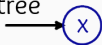


Un modèle plus complexe : la régression quadratique

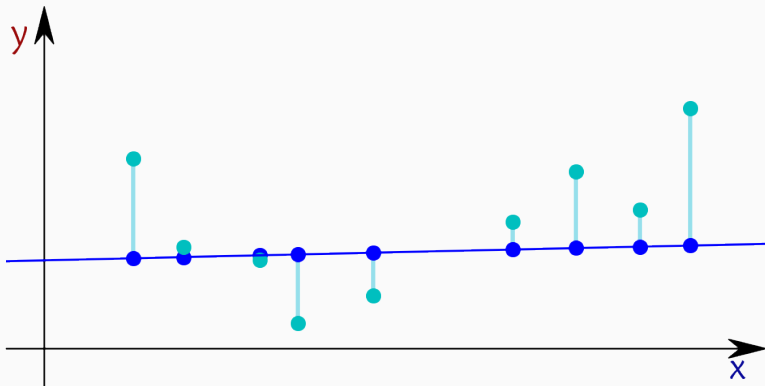
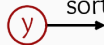
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

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

entrée



sortie

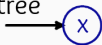


Un modèle plus complexe : la régression quadratique

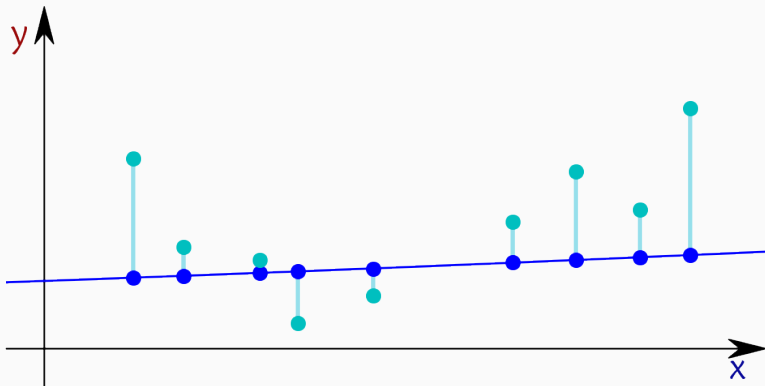
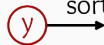
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +0.00, +0.04, +0.09$$

entrée



sortie



Un modèle plus complexe : la régression quadratique

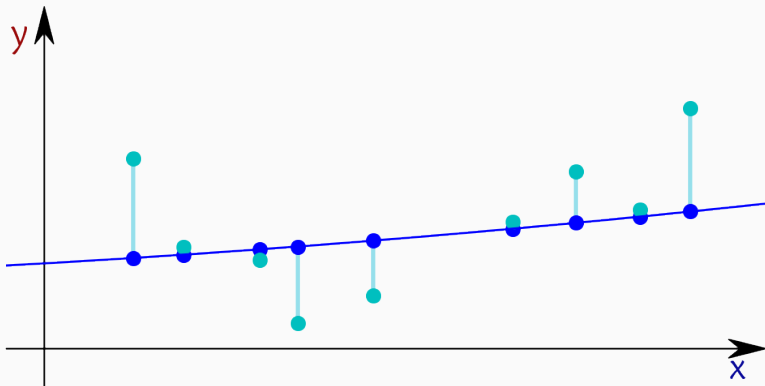
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +0.03, +0.08, +0.11$$

entrée



sortie



Un modèle plus complexe : la régression quadratique

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

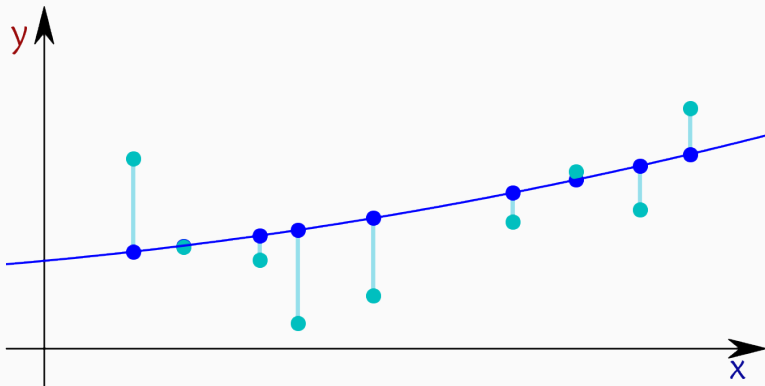
$$(a, b, c) = +0.08, +0.17, +0.12$$

entrée

x

y

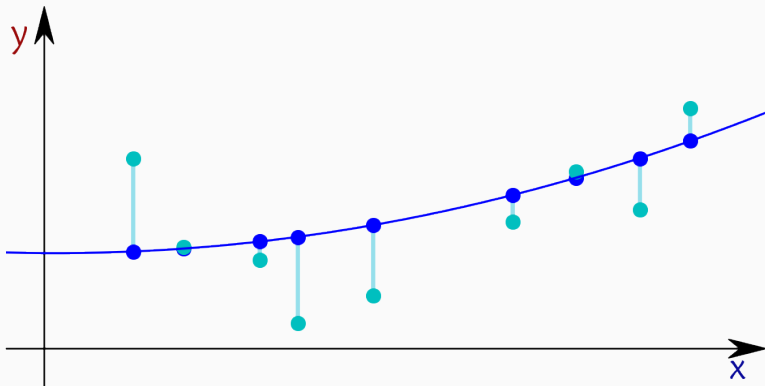
sortie



Un modèle plus complexe : la régression quadratique

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

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

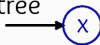


Un modèle plus complexe : la régression quadratique

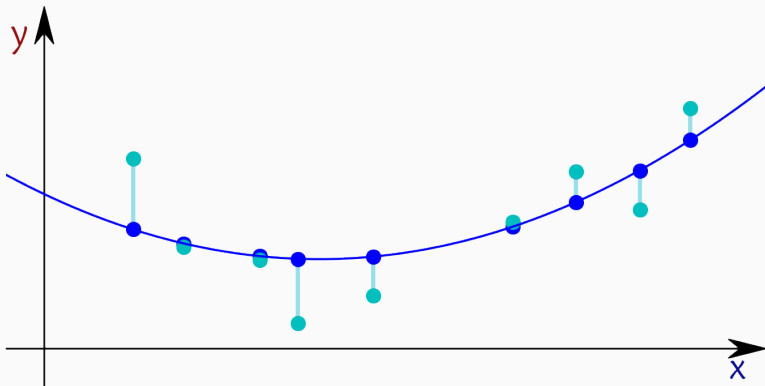
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

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

entrée



sortie

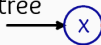


Un modèle plus complexe : la régression quadratique

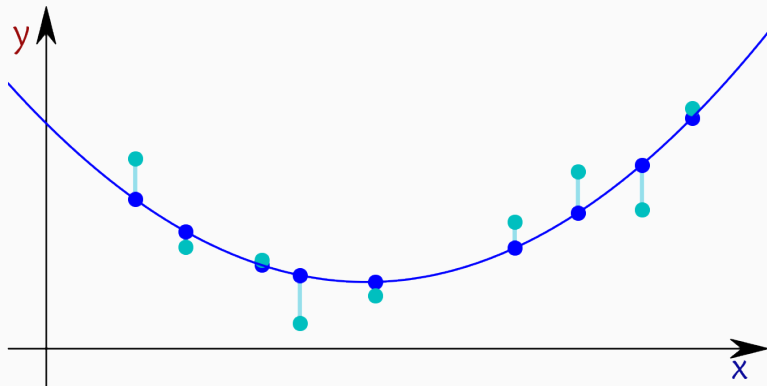
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +1.18, +0.07, +0.30$$

entrée



sortie

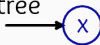


Un modèle plus complexe : la régression quadratique

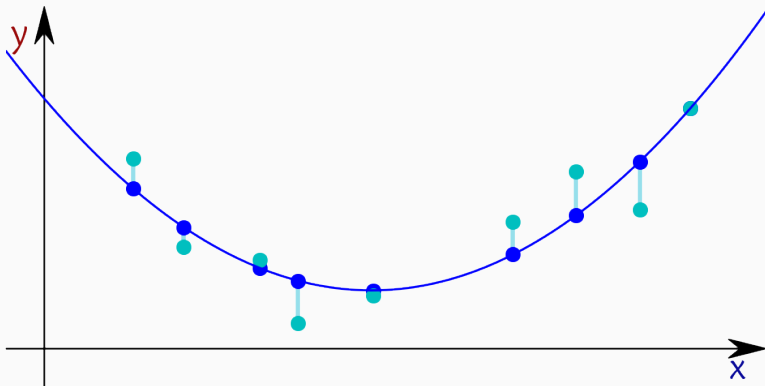
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +1.36, +0.05, +0.33$$

entrée



sortie

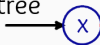


Un modèle plus complexe : la régression quadratique

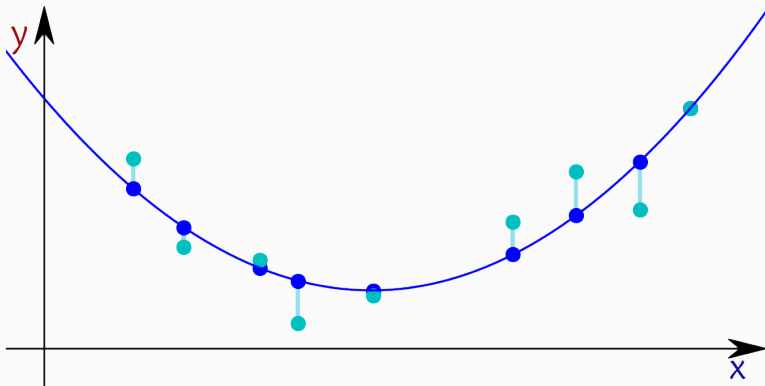
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +1.36, +0.05, +0.33$$

entrée



sortie

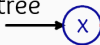


Un modèle plus complexe : la régression quadratique

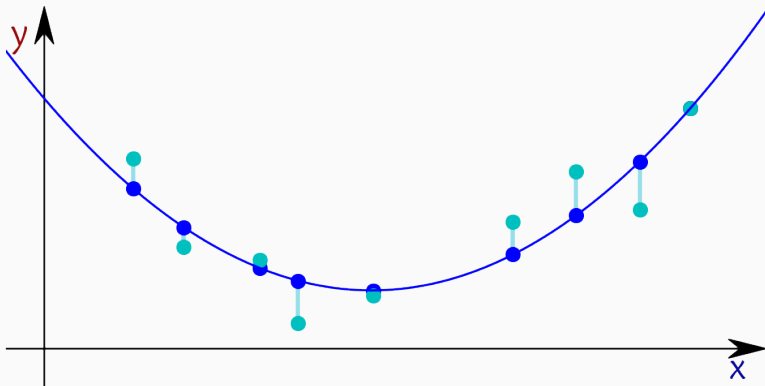
$$F(a, b, c; x) = a \cdot x^2 + b \cdot x + c$$

$$(a, b, c) = +1.36, +0.05, +0.33$$

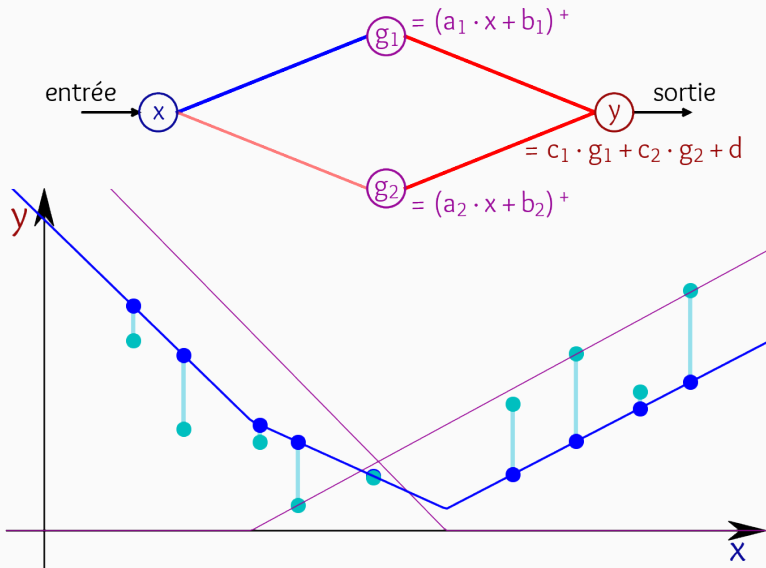
entrée



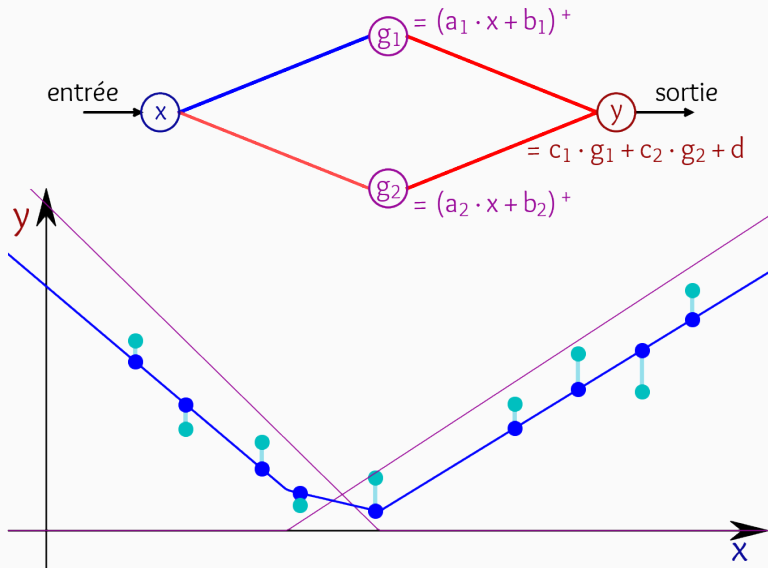
sortie



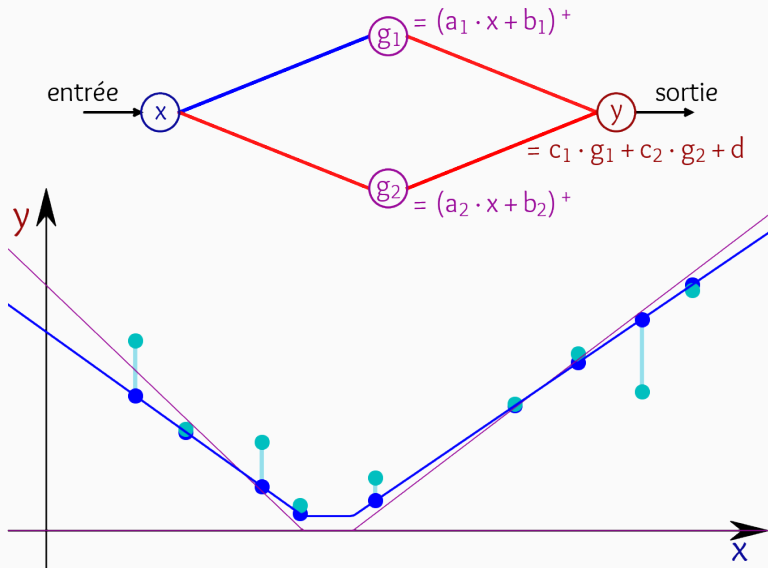
Complexifions le modèle par l'ajout de variables intermédiaires...



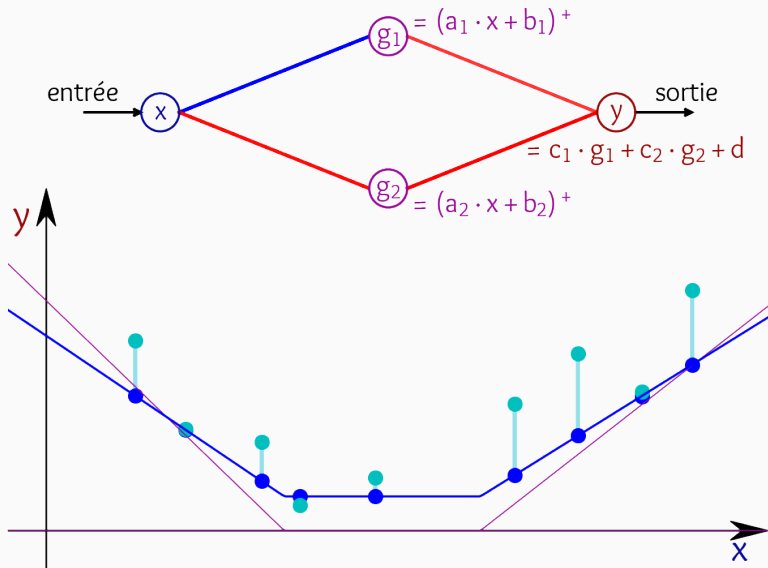
Complexifions le modèle par l'ajout de variables intermédiaires...



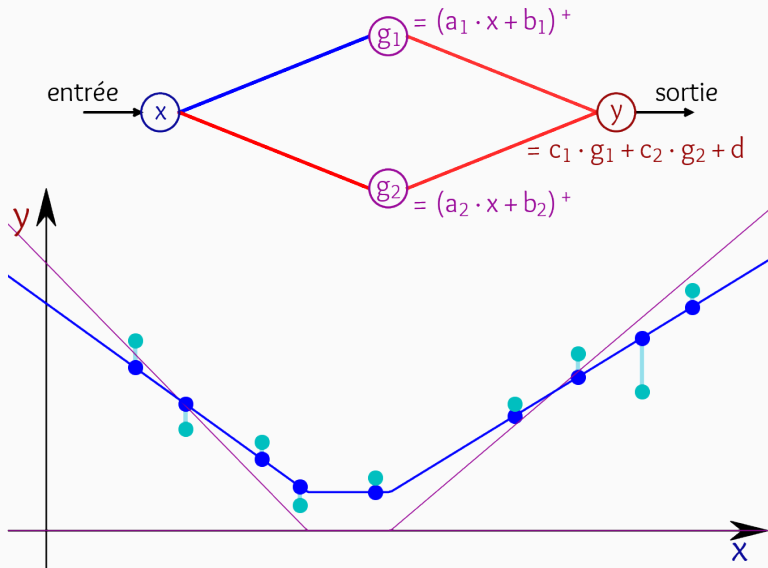
Complexifions le modèle par l'ajout de variables intermédiaires...



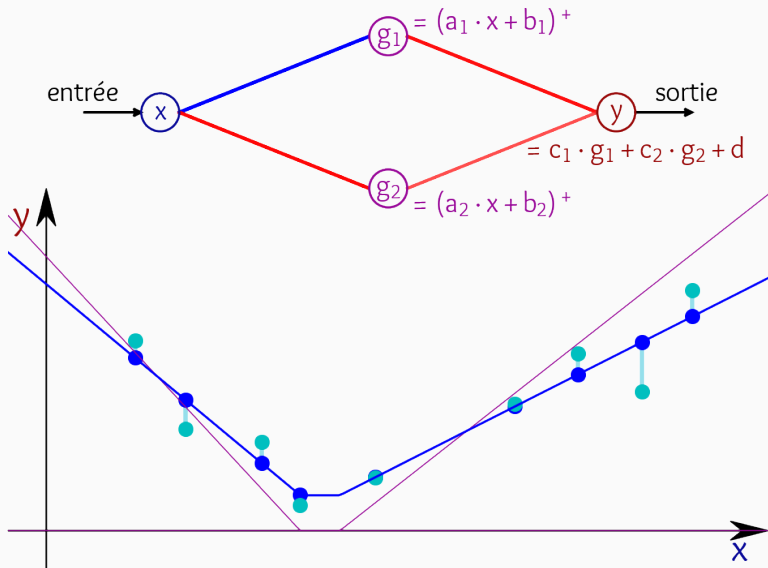
Complexifions le modèle par l'ajout de variables intermédiaires...



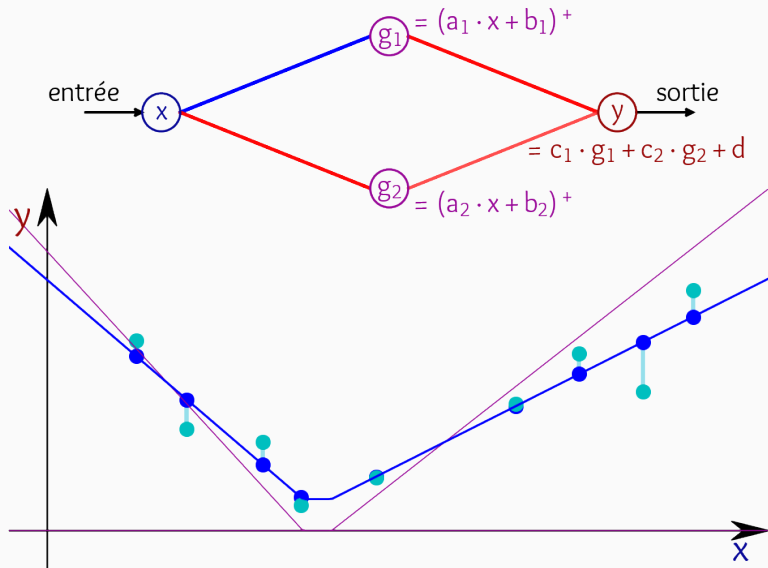
Complexifions le modèle par l'ajout de variables intermédiaires...



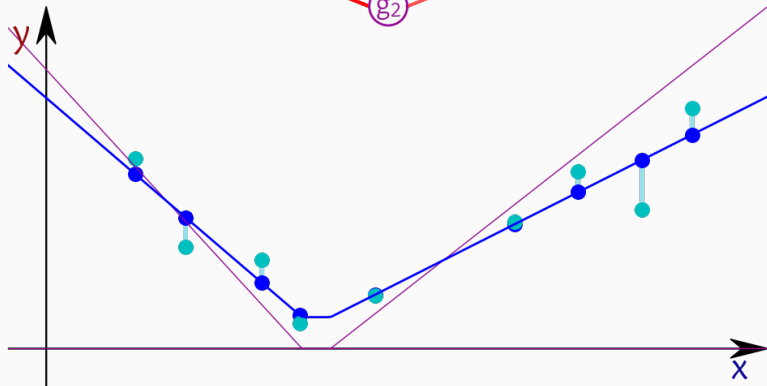
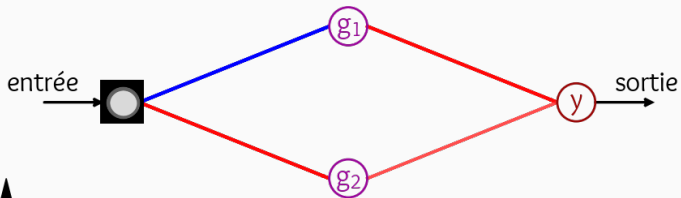
Complexifions le modèle par l'ajout de variables intermédiaires...



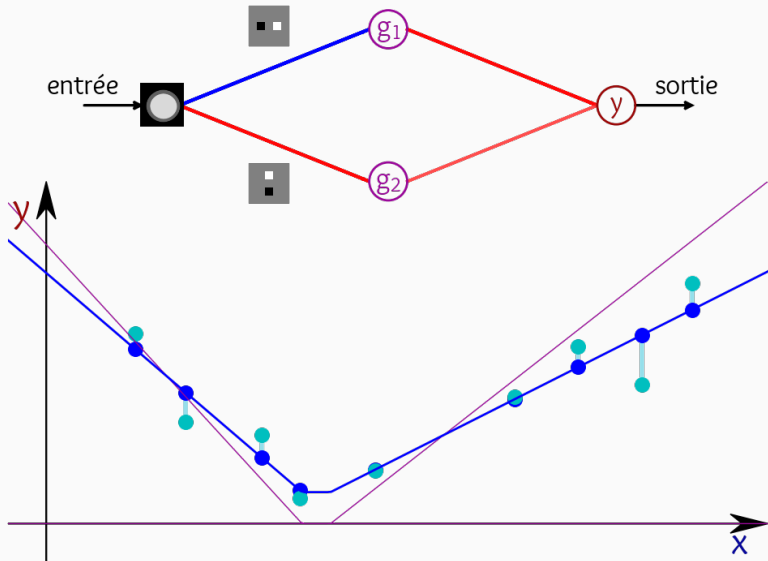
Complexifions le modèle par l'ajout de variables intermédiaires...



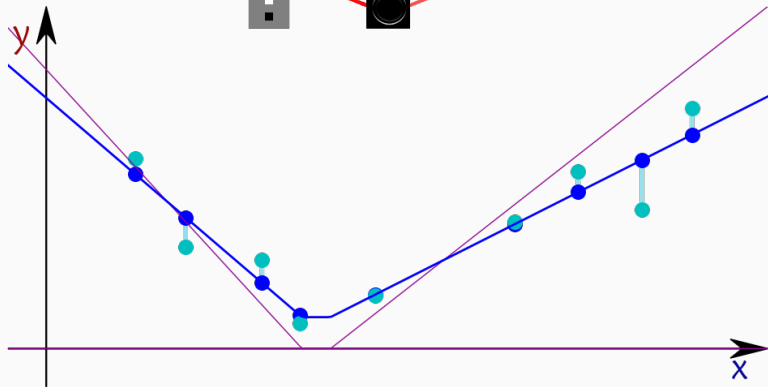
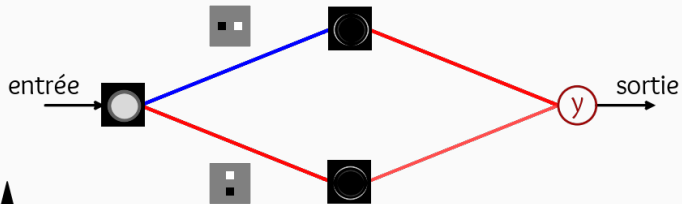
Complexifions le modèle par l'ajout de variables intermédiaires...



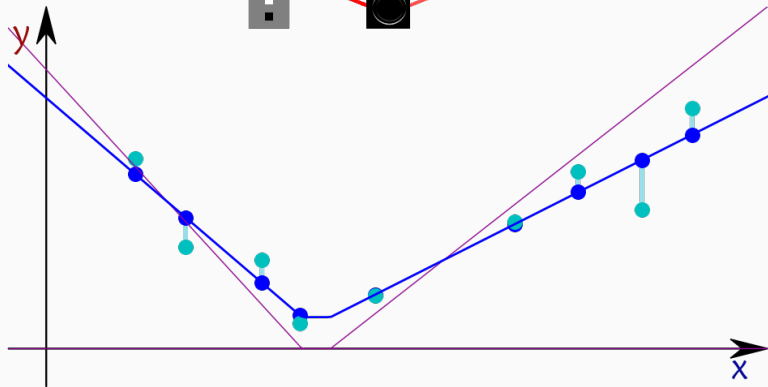
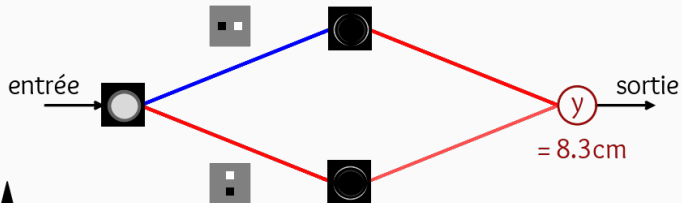
Complexifions le modèle par l'ajout de variables intermédiaires...



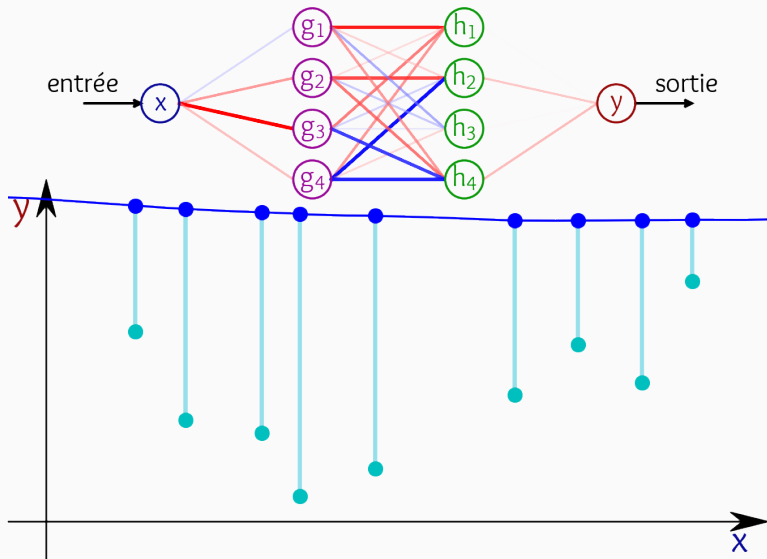
Complexifions le modèle par l'ajout de variables intermédiaires...



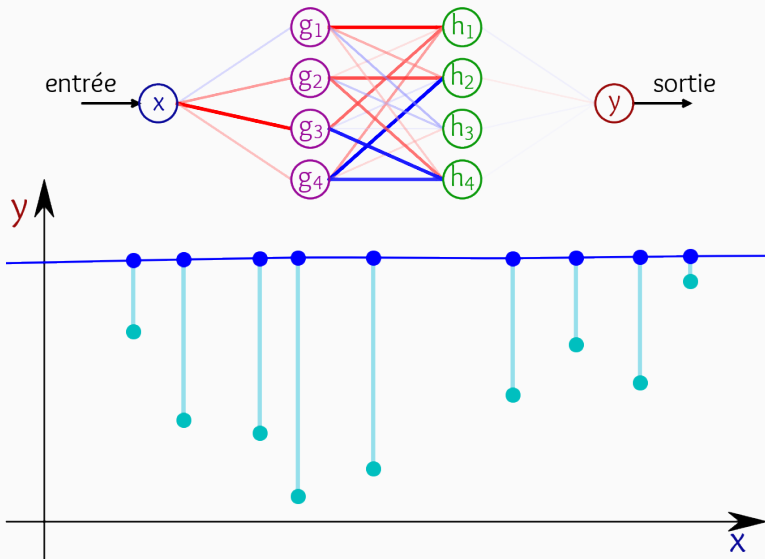
Complexifions le modèle par l'ajout de variables intermédiaires...



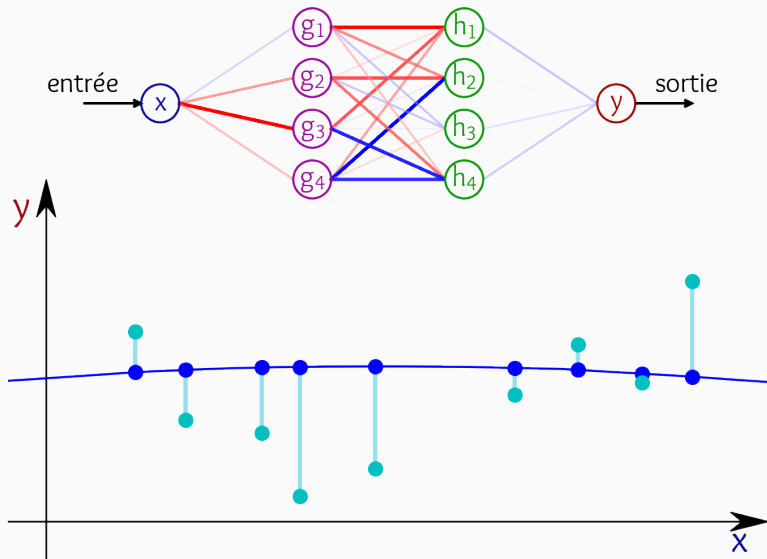
On peut alors augmenter le nombre de “neurones” et de couches !



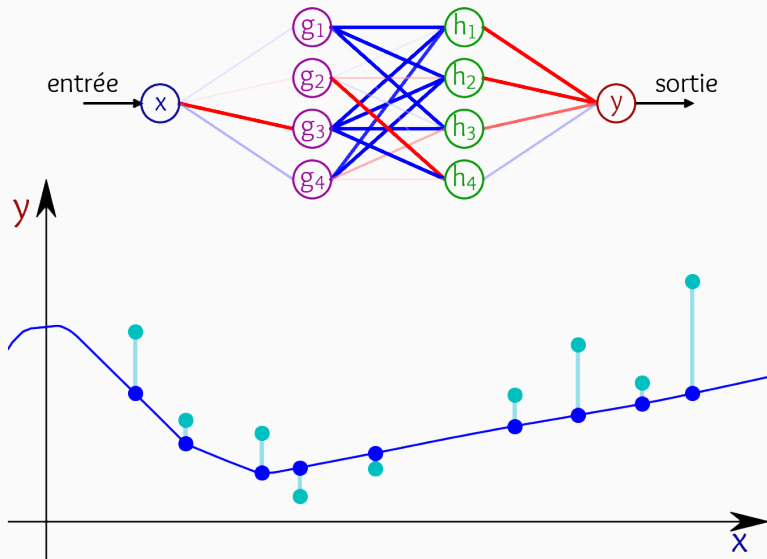
On peut alors augmenter le nombre de “neurones” et de couches !



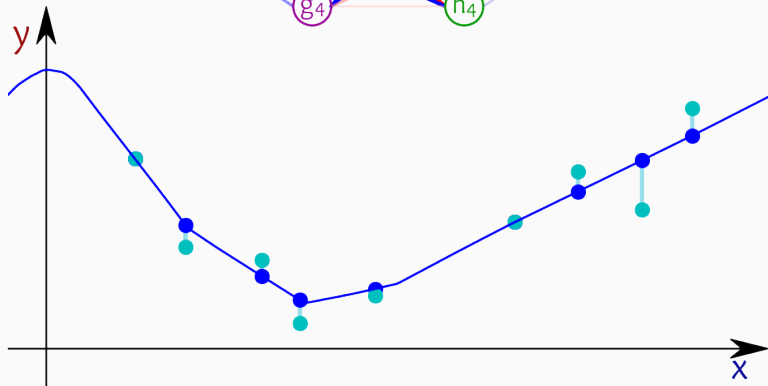
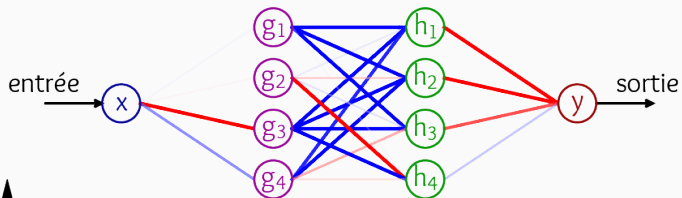
On peut alors augmenter le nombre de “neurones” et de couches !



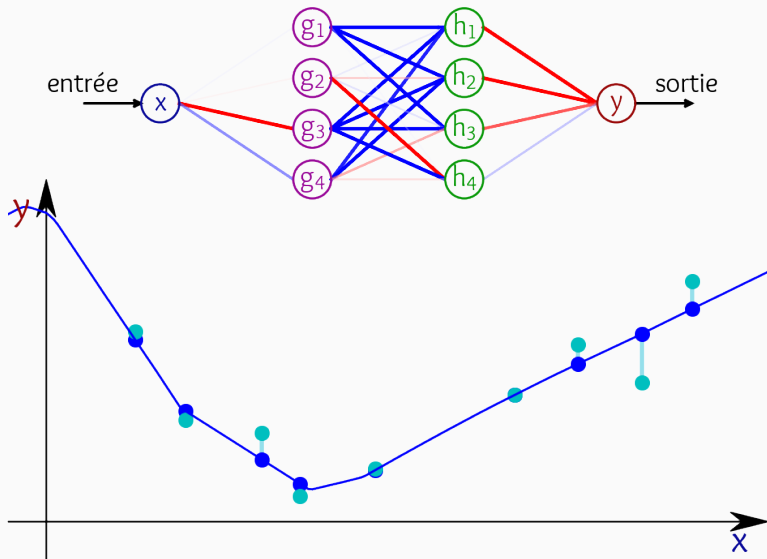
On peut alors augmenter le nombre de “neurones” et de couches!



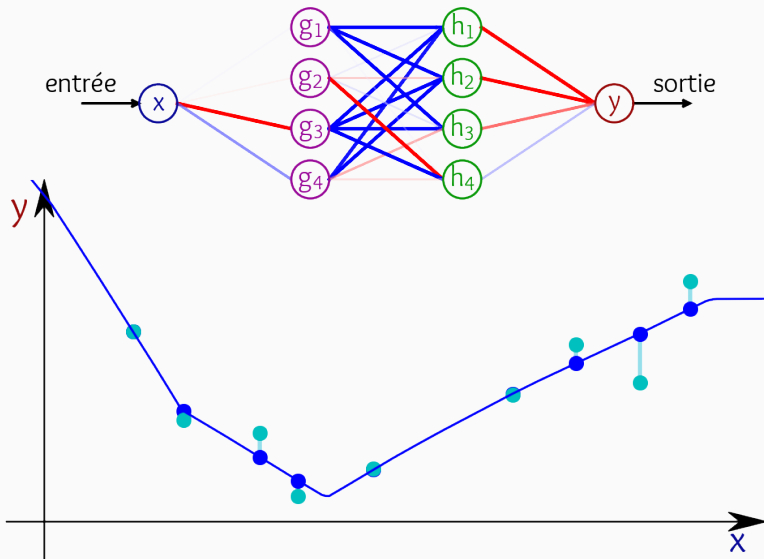
On peut alors augmenter le nombre de “neurones” et de couches!



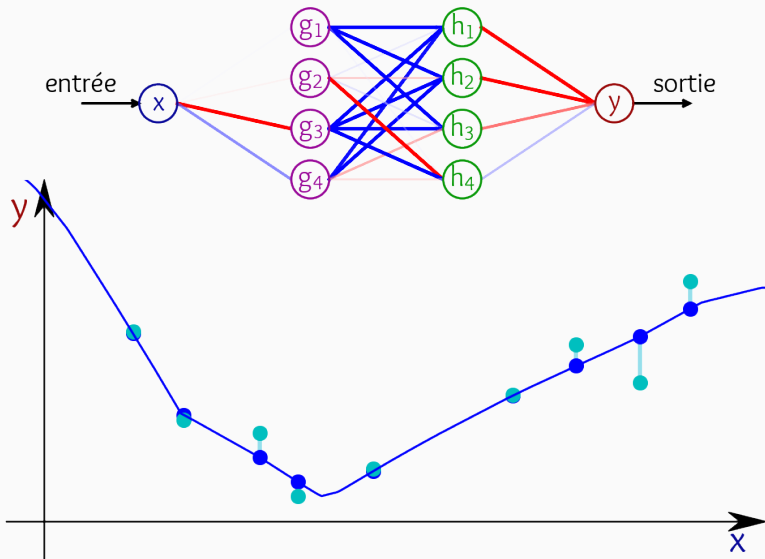
On peut alors augmenter le nombre de “neurones” et de couches!



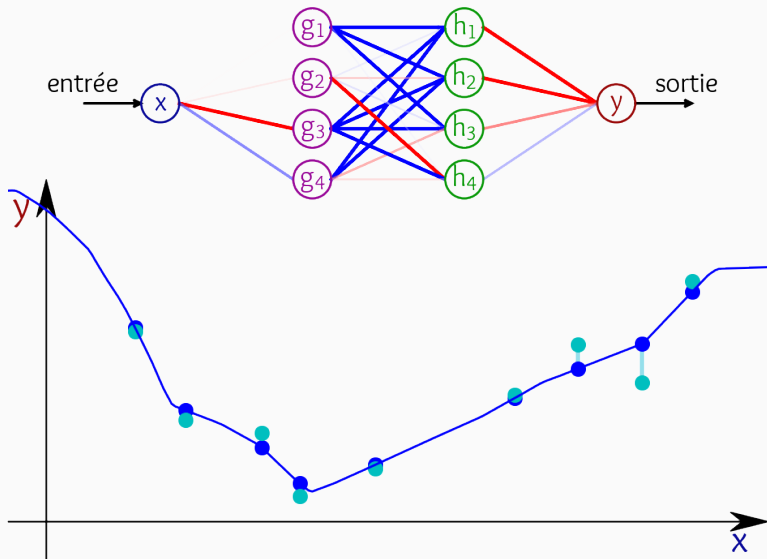
On peut alors augmenter le nombre de “neurones” et de couches!



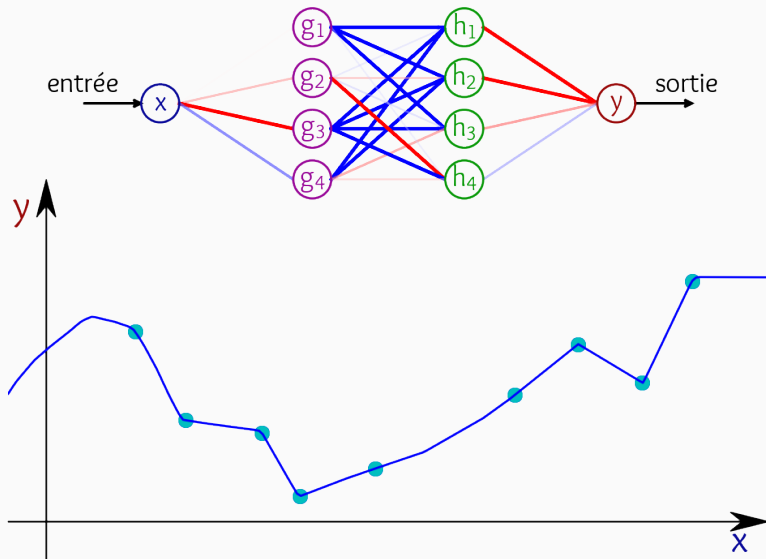
On peut alors augmenter le nombre de “neurones” et de couches!



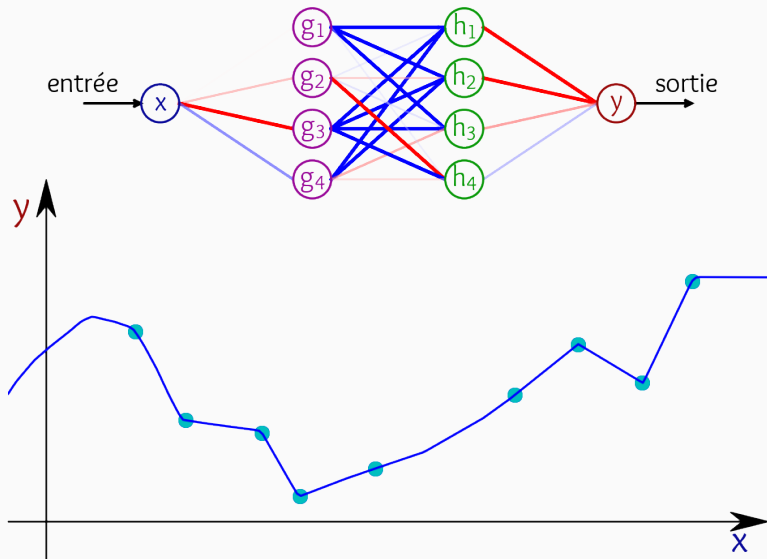
On peut alors augmenter le nombre de “neurones” et de couches!



On peut alors augmenter le nombre de “neurones” et de couches!



On peut alors augmenter le nombre de “neurones” et de couches!



- Généralisation de la **régression linéaire** à des modèles arbitraires.

- Généralisation de la **régression linéaire** à des modèles arbitraires.
- Introduction de **variables intermédiaires** et de **coudes**.

- Généralisation de la **régression linéaire** à des modèles arbitraires.
- Introduction de **variables intermédiaires** et de **coudes**.
- Entraînement **naïf** (barre souple + ressorts).

- Généralisation de la **régression linéaire** à des modèles arbitraires.
- Introduction de **variables intermédiaires** et de **coudes**.
- Entraînement **naïf** (barre souple + ressorts).

- Généralisation de la **régression linéaire** à des modèles arbitraires.
- Introduction de **variables intermédiaires** et de **coudes**.
- Entraînement **naïf** (barre souple + ressorts).

Réseau de neurones générique
 \simeq **Interpolation** entre les échantillons de la BDD.

- Généralisation de la **régression linéaire** à des modèles arbitraires.
- Introduction de **variables intermédiaires** et de **coudes**.
- Entraînement **naïf** (barre souple + ressorts).

Réseau de neurones générique
 \simeq **Interpolation** entre les échantillons de la BDD.

Est-ce suffisant?

Analyse d'images \neq Big Data générique

Analyse d'images \neq Big Data générique



1 nombre

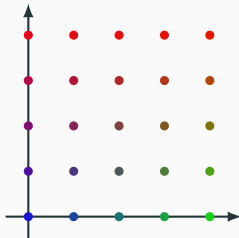
→ 5 échantillons

Analyse d'images \neq Big Data générique



1 nombre

→ 5 échantillons



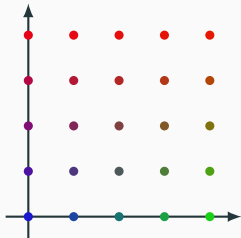
2 nombres

→ 5^2 échantillons

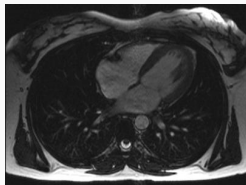
Analyse d'images \neq Big Data générique



1 nombre
→ 5 échantillons



2 nombres
→ 5^2 échantillons

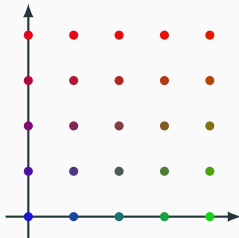


$128 \cdot 128$ nombres
→ $5^{128 \cdot 128}$ échantillons



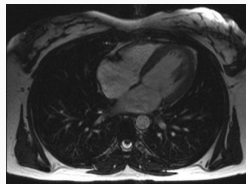
1 nombre

→ 5 échantillons



2 nombres

→ 5^2 échantillons



128 · 128 nombres

→ $5^{128 \cdot 128}$ échantillons

L'ensemble de toutes les images est **beaucoup trop vaste** pour pouvoir être échantillonné avec la précision requise.

En imagerie médicale, un bon modèle $F(\mathbf{w}; x) \simeq y$ doit donc :

Refléter un **a priori** raisonnable sur les données

Qu'est-ce qu'un scan IRM de cœur? Un modèle déformé?

En imagerie médicale, un bon modèle $F(\mathbf{w}; x) \simeq y$ doit donc :

Refléter un **a priori** raisonnable sur les données

Qu'est-ce qu'un scan IRM de cœur? Un modèle déformé?

Être **contraignant**

Pour guider le modèle **en dehors** des cas couverts par la BDD.

En imagerie médicale, un bon modèle $F(\mathbf{w}; x) \simeq y$ doit donc :

Refléter un **a priori** raisonnable sur les données

Qu'est-ce qu'un scan IRM de cœur? Un modèle déformé?

Être **contraignant**

Pour guider le modèle **en dehors** des cas couverts par la BDD.

Reposer sur des opérations **élémentaires**

Pour tourner sur des volumes 3D/4D.

En imagerie médicale, un bon modèle $F(\mathbf{w}; x) \simeq y$ doit donc :

Refléter un **a priori** raisonnable sur les données

Qu'est-ce qu'un scan IRM de cœur ? Un modèle déformé ?

Être **contraignant**

Pour guider le modèle **en dehors** des cas couverts par la BDD.

Reposer sur des opérations **élémentaires**

Pour tourner sur des volumes 3D/4D.

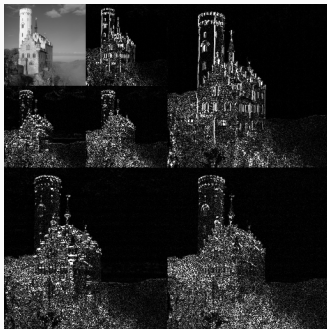
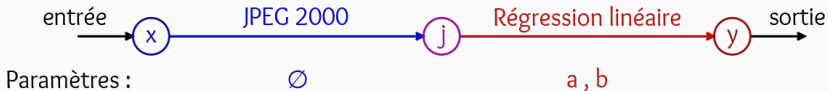
\implies **Combinons la régression avec JPEG2000!**

Analyse par filtrage multi-résolution :

⇒ Réseaux de neurones convolutifs

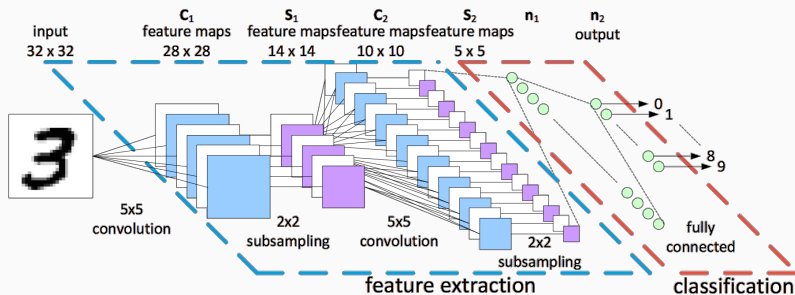
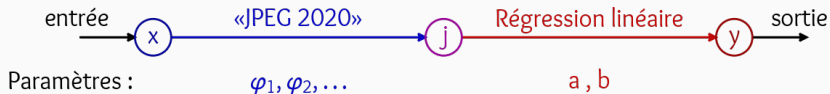
Vers une optimisation des filtres de JPEG 2000

Traitement du signal classique [Dam] :



Vers une optimisation des filtres de JPEG 2000

Traitement du signal moderne [PMC11] :



Convolutional Neural Networks : un excellent compromis

JPEG2000, c'est un modèle $F(\mathbf{w}; \mathbf{x}) \simeq \mathbf{y}$:

- Algorithmiquement simple.

Convolutional Neural Networks : un excellent compromis

JPEG2000, c'est un modèle $F(\mathbf{w}; \mathbf{x}) \simeq \mathbf{y}$:

- Algorithmiquement simple.
- Contraignant.

Convolutional Neural Networks : un excellent compromis

JPEG2000, c'est un modèle $F(\mathbf{w}; \mathbf{x}) \simeq \mathbf{y}$:

- Algorithmiquement simple.
- Contraignant.
- Encodant un a priori **multi-résolution** sur les images.

Convolutional Neural Networks : un excellent compromis

JPEG2000, c'est un modèle $F(\mathbf{w}; \mathbf{x}) \simeq \mathbf{y}$:

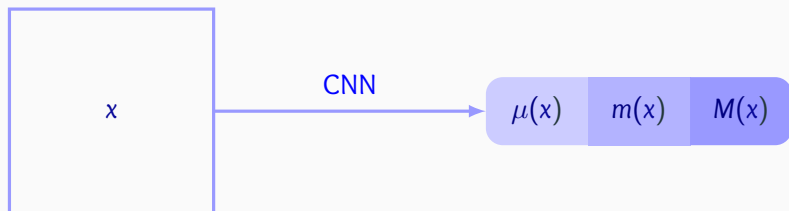
- Algorithmiquement simple.
- Contraignant.
- Encodant un a priori **multi-résolution** sur les images.

Convolutional Neural Networks : un excellent compromis

JPEG2000, c'est un modèle $F(\mathbf{w}; x) \simeq y$:

- Algorithmiquement simple.
- Contraignant.
- Encodant un a priori **multi-résolution** sur les images.

En **optimisant ses coefficients** sur une base d'images labellisées, on obtient un **CNN** \simeq "JPEG 2020" adapté au problème.



Une application emblématique : le Deep Art [NN16]

Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



μ m M

μ m M



Une application emblématique : le Deep Art [NN16]



μ

m

M

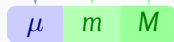
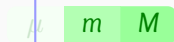
μ

m

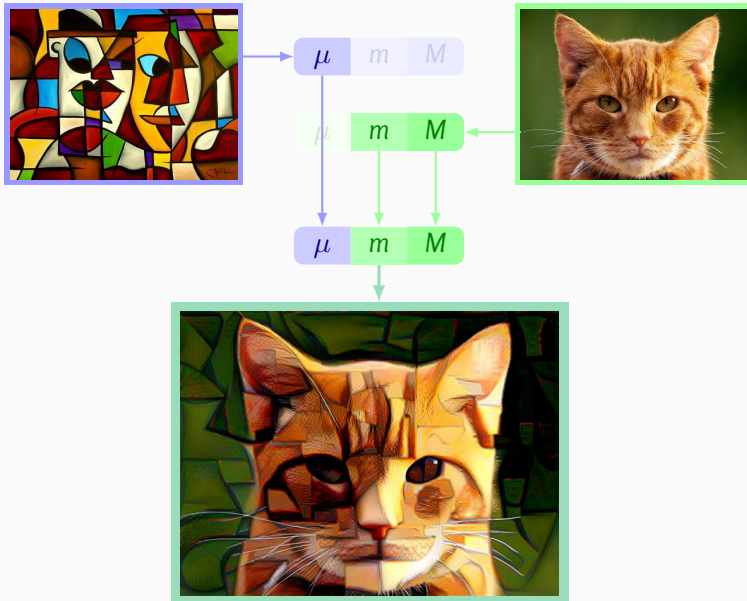
M



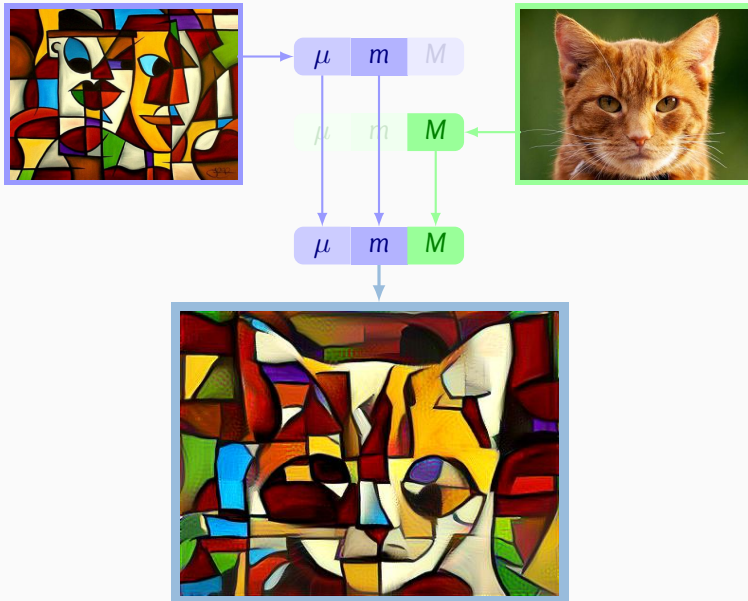
Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



μ m M

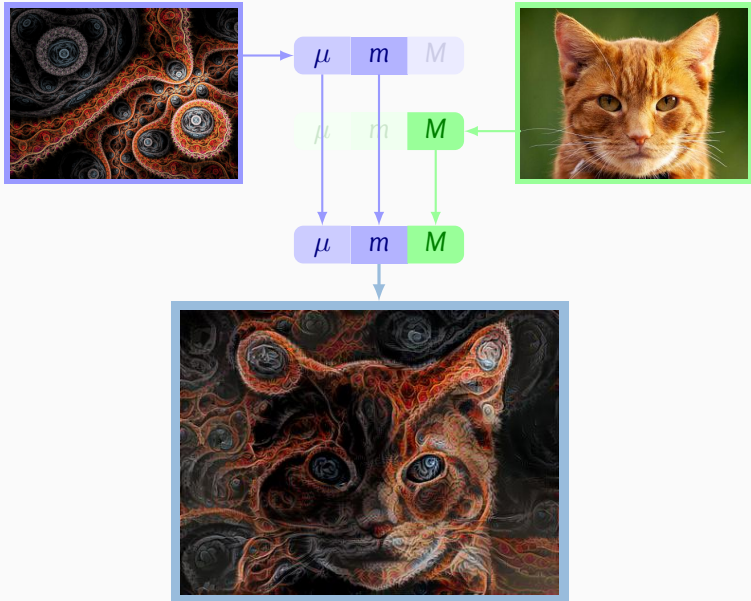


μ m M

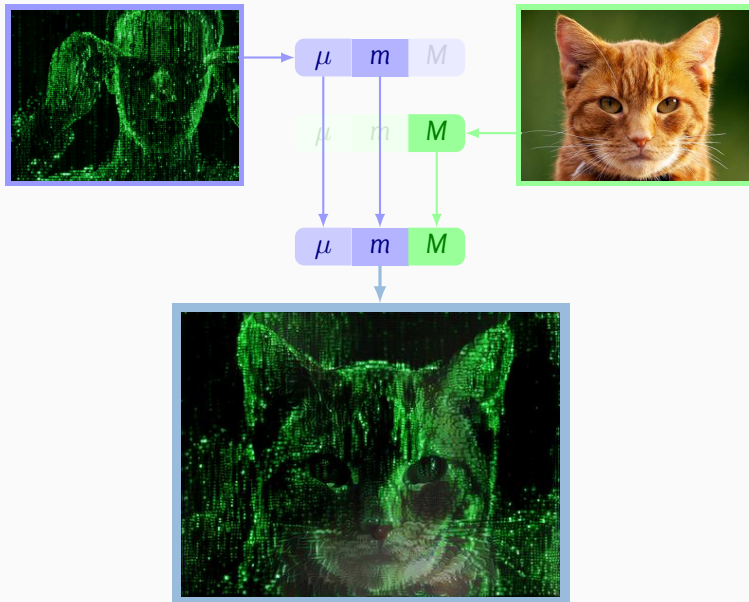
μ m M



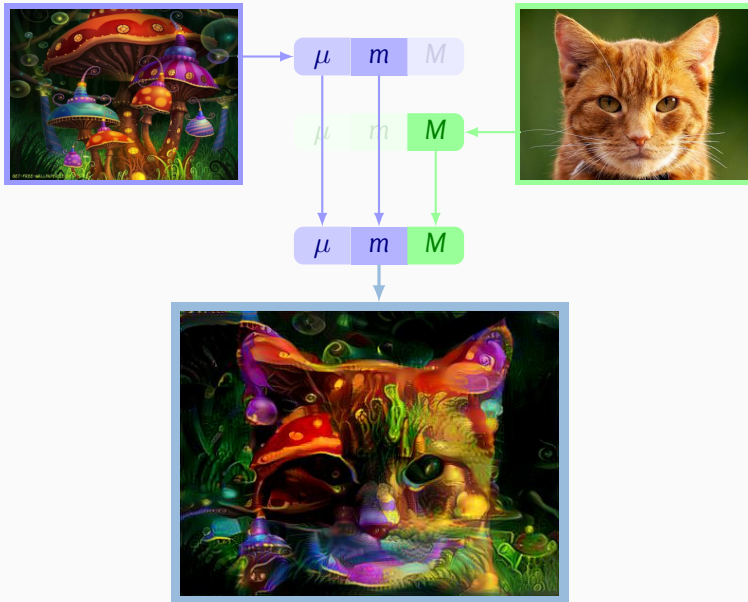
Une application emblématique : le Deep Art [NN16]



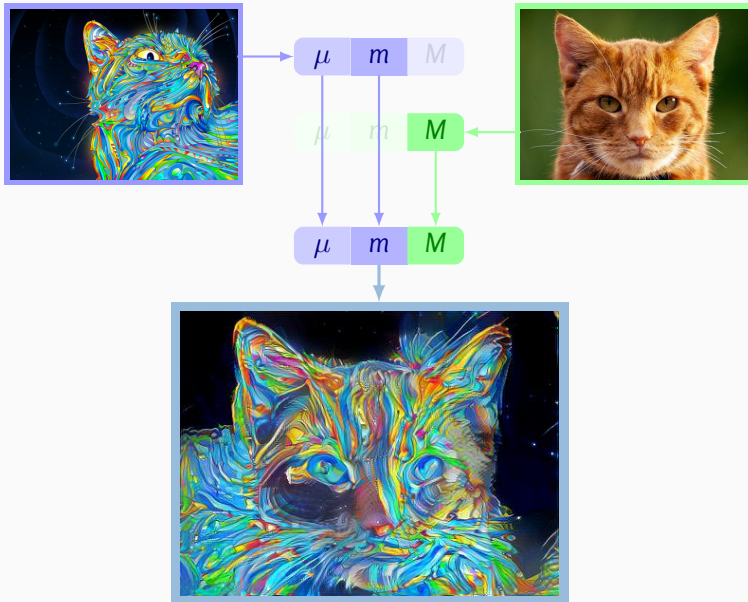
Une application emblématique : le Deep Art [NN16]



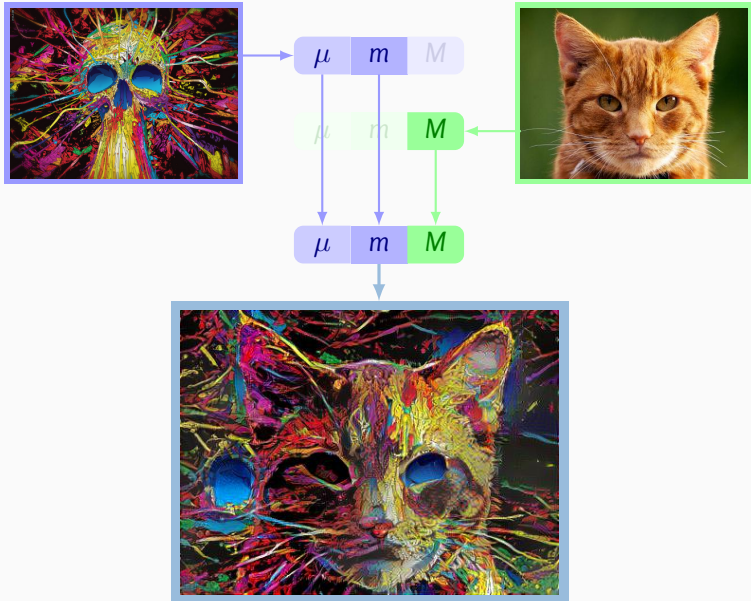
Une application emblématique : le Deep Art [NN16]



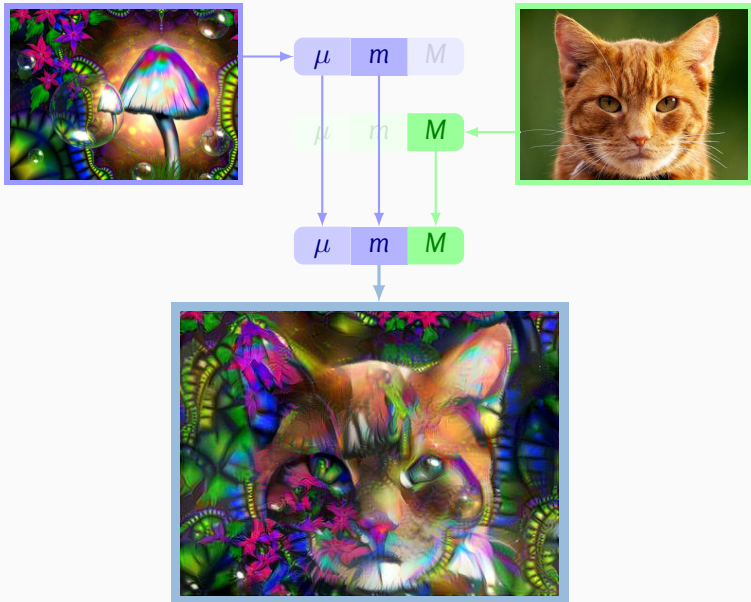
Une application emblématique : le Deep Art [NN16]



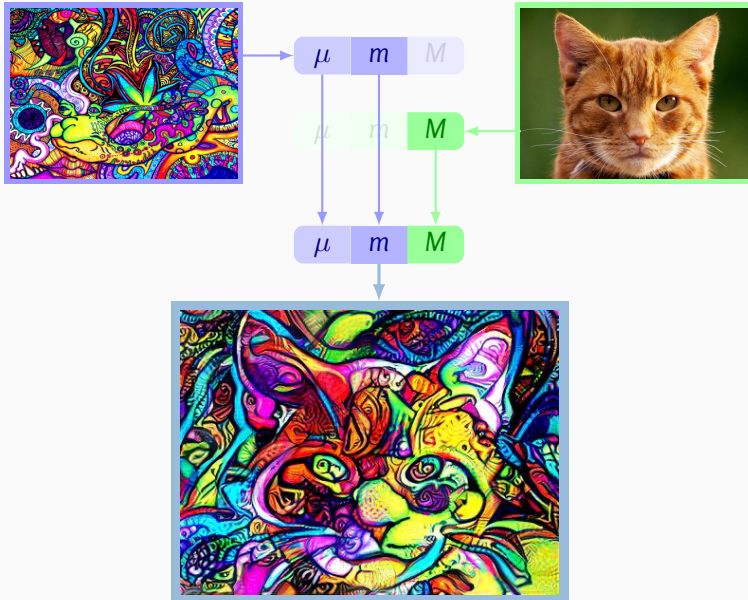
Une application emblématique : le Deep Art [NN16]



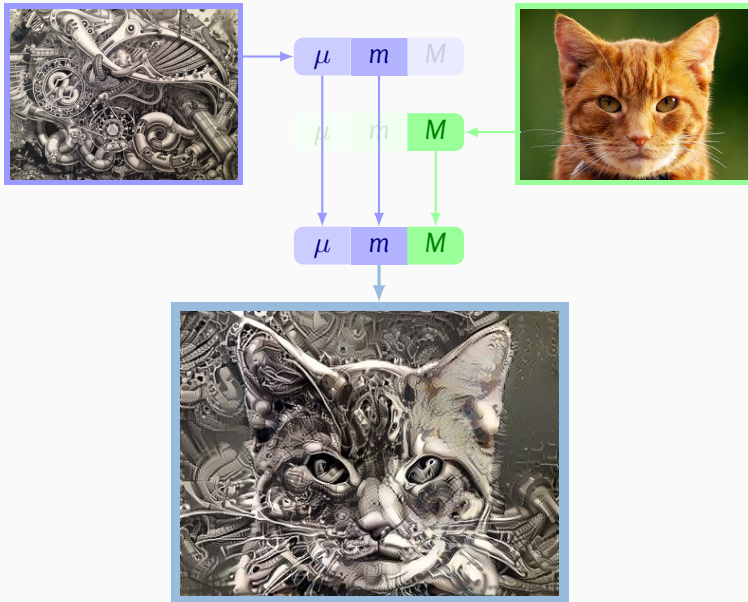
Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



Une application emblématique : le Deep Art [NN16]



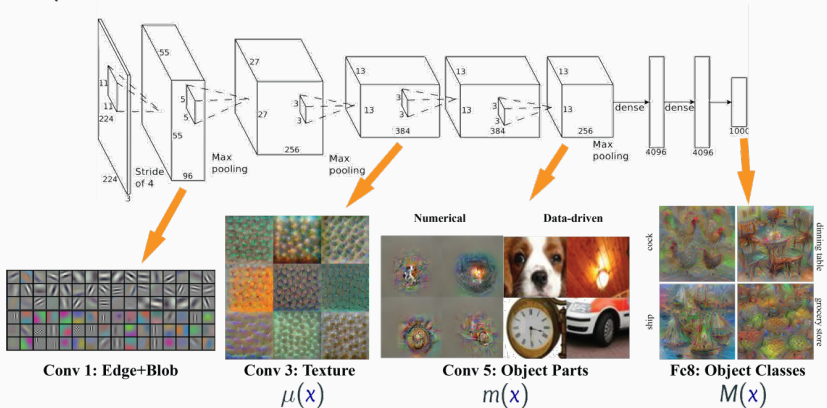
Une application rêvée : la classification d'images

À partir de $\text{CNN}(x) = [\mu(x), m(x), M(x)]$,
peut-on facilement **distinguer** une mouette d'un panda?

Une application rêvée : la classification d'images

À partir de $\text{CNN}(x) = [\mu(x), m(x), M(x)]$,
peut-on facilement **distinguer** une mouette d'un panda?

Ce que les chercheurs ont en tête [WZTF] :



Limites de l'analyse multi-résolution

Les CNNs font de la **détection de motif**, ni plus ni moins [NYC15] :

Limites de l'analyse multi-résolution

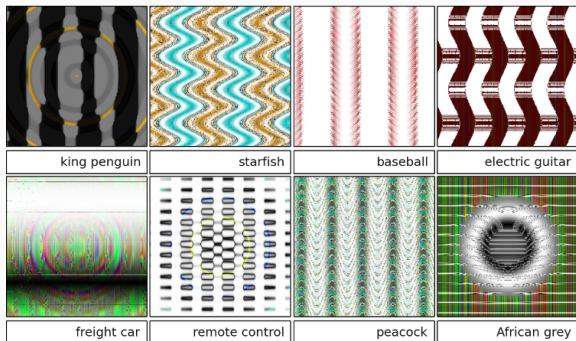
Les CNNs font de la **détection de motif**, ni plus ni moins [NYC15] :

« $\mu(x)$ est fiable; $M(x)$ ne l'est pas. »

Limites de l'analyse multi-résolution

Les CNNs font de la **détection de motif**, ni plus ni moins [NYC15] :

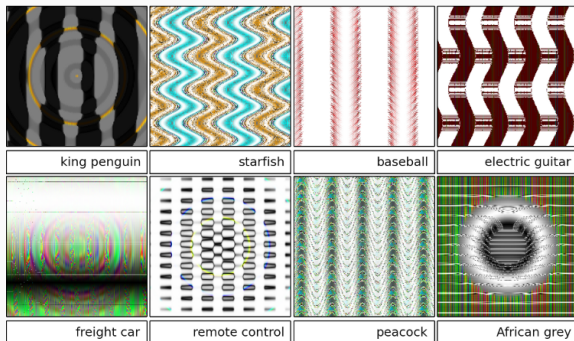
« $\mu(x)$ est fiable; $M(x)$ ne l'est pas. »



Limites de l'analyse multi-résolution

Les CNNs font de la **détection de motif**, ni plus ni moins [NYC15] :

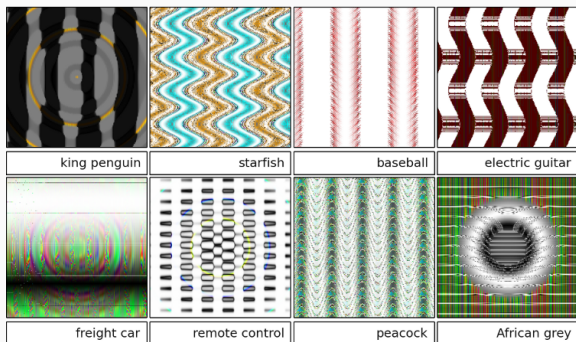
« $\mu(x)$ est fiable; $M(x)$ ne l'est pas. »



Limites de l'analyse multi-résolution

Les CNNs font de la **détection de motif**, ni plus ni moins [NYC15] :

« $\mu(x)$ est fiable; $M(x)$ ne l'est pas. »



Problème : les modèles **structuraux** sont **beaucoup** plus coûteux.
(c'est mon métier...)

Conclusion

Un point sur le vocabulaire

On l'a vu :

Perceptron multicouche



Réseau de neurones

Un point sur le vocabulaire

On l'a vu :

Perceptron multicouche



Réseau de neurones

Régression sur une banque
de filtres multi-résolution



Apprentissage profond

Un point sur le vocabulaire

On l'a vu :

Perceptron multicouche

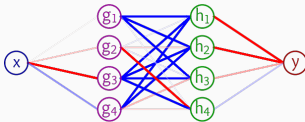


Réseau de neurones

Régression sur une banque
de filtres multi-résolution



Apprentissage profond



Un point sur le vocabulaire

On l'a vu :

Perceptron multicouche

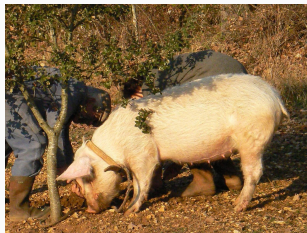


Réseau de neurones

Régression sur une banque
de filtres multi-résolution



Apprentissage profond



La révolution du Deep Learning, c'est :

- L'intelligence artificielle.

La révolution du Deep Learning, c'est :

- L'intelligence artificielle.
- Le premier modèle de **texture** vraiment convaincant.

La révolution du Deep Learning, c'est :

- L'intelligence artificielle.
- Le premier modèle de **texture** vraiment convaincant.
- Le développement d'**outils logiciels** permettant de **tuner les paramètres** de nos modèles – TensorFlow, PyTorch...

La révolution du Deep Learning, c'est :

- L'intelligence artificielle.
- Le premier modèle de **texture** vraiment convaincant.
- Le développement d'**outils logiciels** permettant de **tuner les paramètres** de nos modèles – TensorFlow, PyTorch...

La révolution du Deep Learning, c'est :

- L'intelligence artificielle.
- Le premier modèle de **texture** vraiment convaincant.
- Le développement d'**outils logiciels** permettant de **tuner les paramètres** de nos modèles – TensorFlow, PyTorch...

Un logiciel d'analyse d'images repose **toujours** sur une **modélisation simpliste** des données.

Il n'y a pas de miracle.

Devant un produit “IA” :

- « Montrez-moi ce qui ne marche pas. »

Devant un produit “IA” :

- « **Montrez-moi ce qui ne marche pas.** »
- « **Expliquez-moi** cette erreur. »

Devant un produit “IA” :

- « **Montrez-moi ce qui ne marche pas.** »
- « **Expliquez-moi** cette erreur. »
- “Les données” **ne résoudront pas** automagiquement le problème.

Devant un produit “IA” :

- « **Montrez-moi ce qui ne marche pas.** »
- « **Expliquez-moi** cette erreur. »
- “Les données” **ne résoudront pas** automagiquement le problème.

Devant un produit “IA” :

- « **Montrez-moi ce qui ne marche pas.** »
- « **Expliquez-moi** cette erreur. »
- “Les données” **ne résoudront pas** automagiquement le problème.

Pour aller plus loin :

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 !

Merci pour votre attention.

Avez-vous des questions ?



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.

References ii



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.



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.