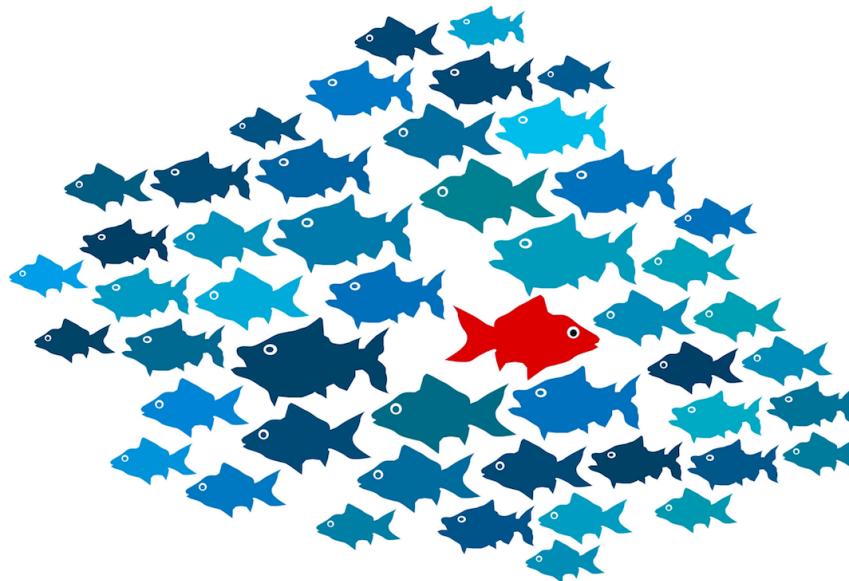


# Détection d'anomalies à l'aide de réseaux de neurones autoencodeurs



11 Décembre 2019 – Campus Cézeaux, UCA



Depuis ~10 ans nouvelle ère pour l'apprentissage machine (\*)

- Facilité d'utilisation des **librairies** de ML (Tensorflow, Keras, Pytorch...)
- **Rapidité** d'exécution des algorithmes : **Graphics Processing Units**
- **Puissance** de calcul : cluster et grilles de calcul
- Nouveaux **algorithmes**: VAE (2013), GAN (2104), ADAM (2014)...
- Algorithmes de plus en plus **complexes** (Deep Learning)
- Très grand nombre **d'applications** scientifiques, industrielles, ...

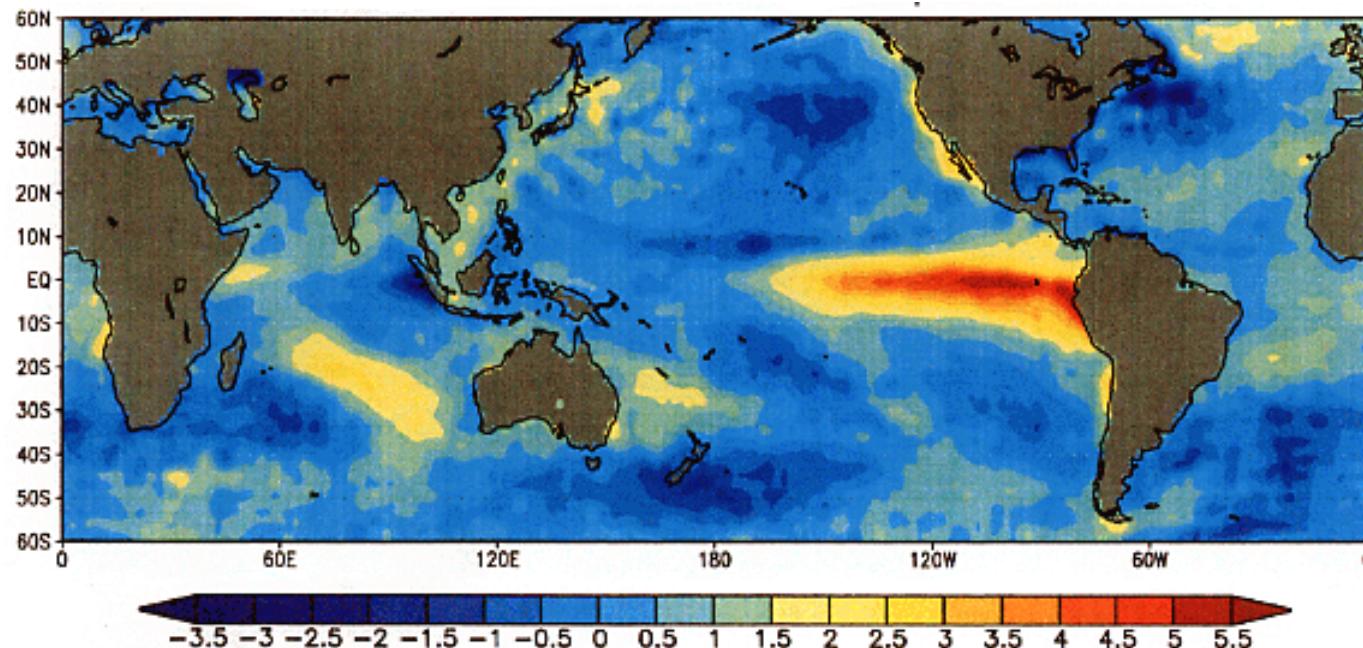
**Détection d'anomalies : détection de phénomènes « non-standard »**

(\*) beaucoup de buzz-words mais « Machine Learning » semble plus adapté

- 1) Anomalies : définitions et exemples
- 2) Machine learning et réseaux de neurones
- 3) Réseaux de neurones Autoencodeurs
- 4) Exemple concret d'utilisation
- 5) Architectures avancées

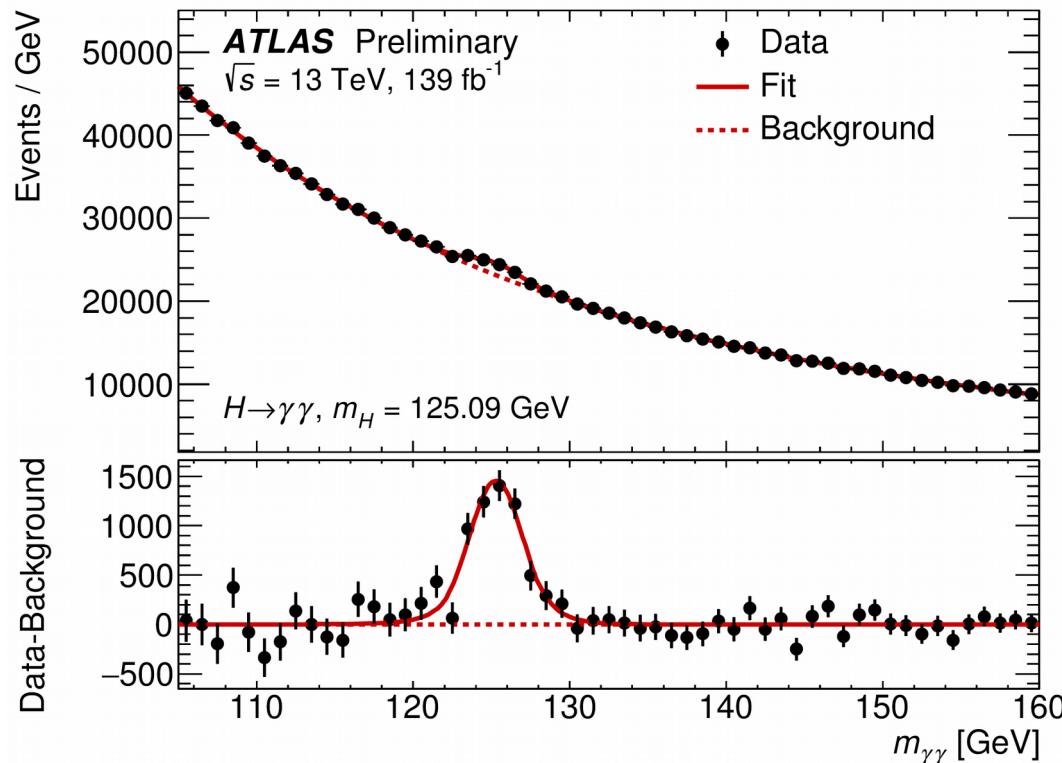
# Anomalies

Anomalies **locales**/ponctuelles (espace/temps) ou **globales** (reproductibles)  
→ Caractérisées par des effets **collectifs**, **isolés** ou **systématiques**



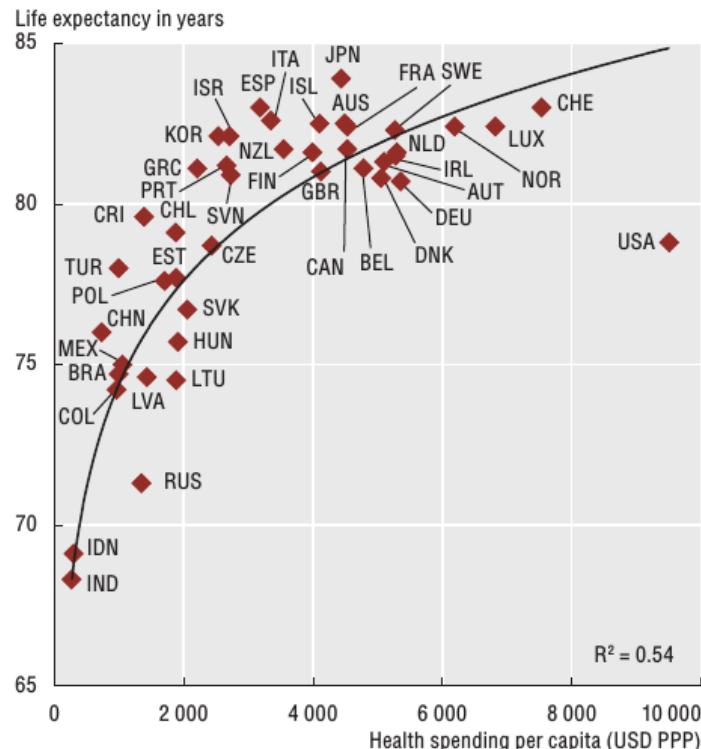
Anomalie locale  
El Niño (12/1997) – wikipedia

Anomalies **locales**/ponctuelles (espace/temps) ou **globales** (reproductibles)  
 → Caractérisées par des effets **collectifs**, **isolés** ou **systématiques**



Anomalie globale (effet collectif)  
 Boson de Higgs - ATLAS

Anomalies **locales**/ponctuelles (espace/temps) ou **globales** (reproductibles)  
→ Caractérisées par des effets **collectifs**, **isolés** ou **systématiques**

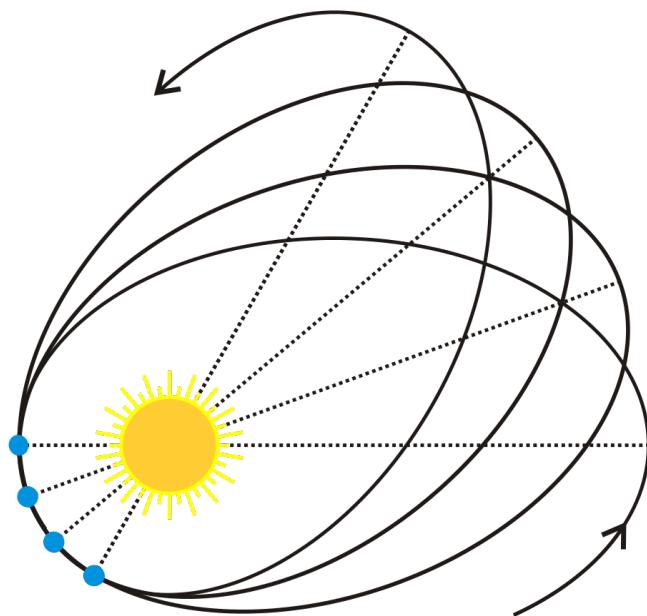


Source: OECD Health Statistics 2017.

StatLink <http://dx.doi.org/10.1787/888933602272>

Données isolées  
Espérance de vie vs dépenses de santé

Anomalies **locales**/ponctuelles (espace/temps) ou **globales** (reproductibles)  
→ Caractérisées par des effets **collectifs**, **isolés** ou **systématiques**

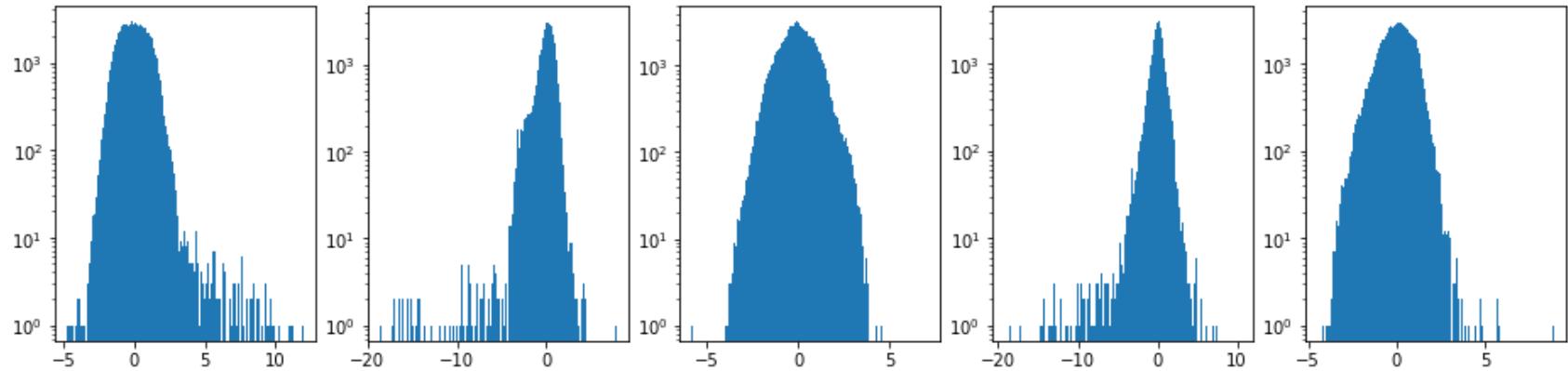


Anomalie de 43"/siècle  
observée depuis le XIX,  
résolue par la Relativité  
Générale

Anomalie systématique  
Précession du périhélie de Mercure

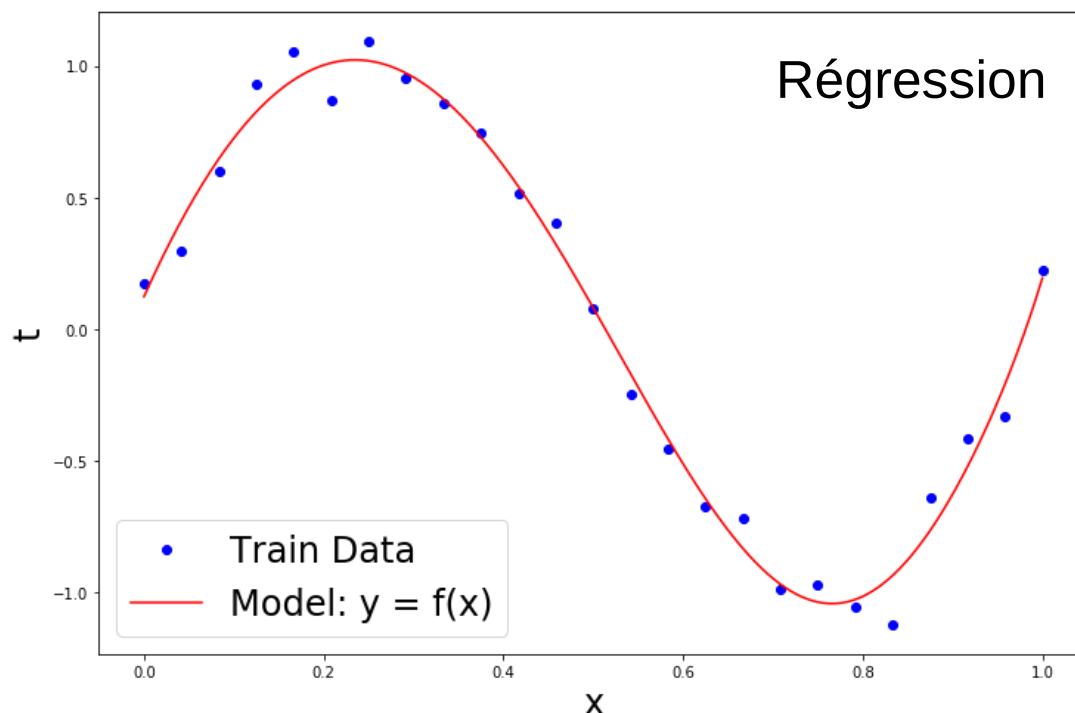
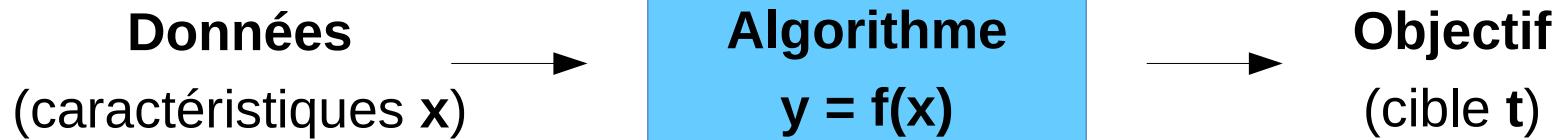
Souvent la **présence** d'anomalies est **difficile à déceler**

- Nécessite d'explorer des **données multidimensionnelles**
- **Approches** statistiques plus **sophistiquées**

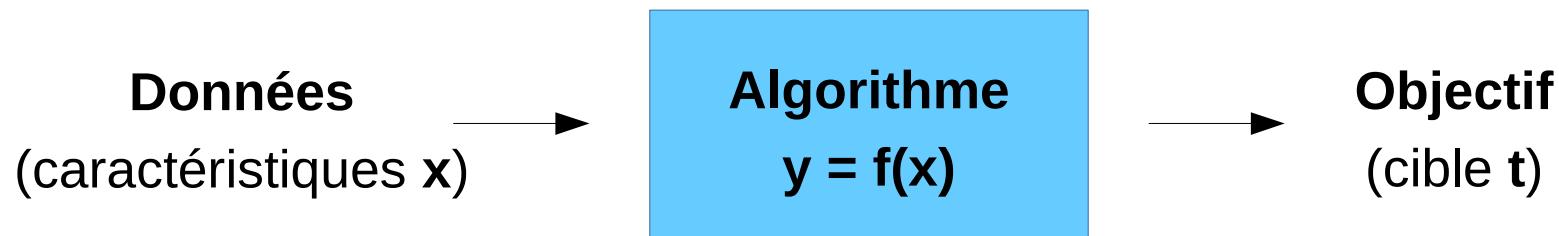


Présence d'anomalies ?

⇒ Méthodes d'apprentissage à **1 seule classe** : **Autoencodeurs**



# Machine Learning

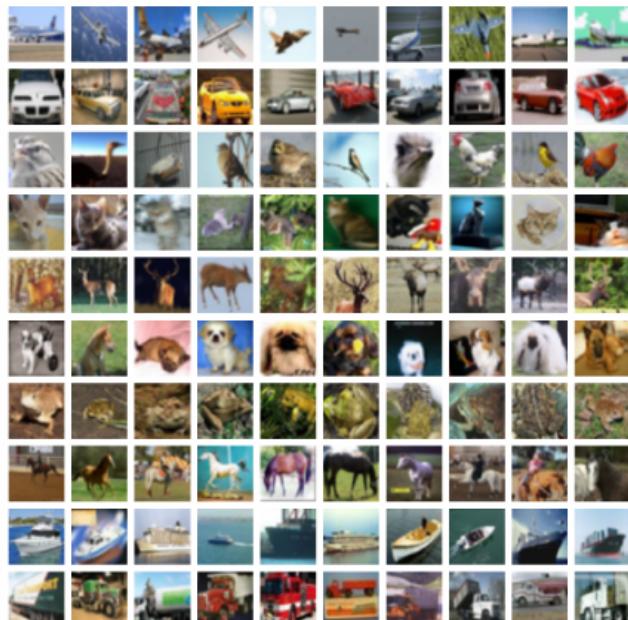
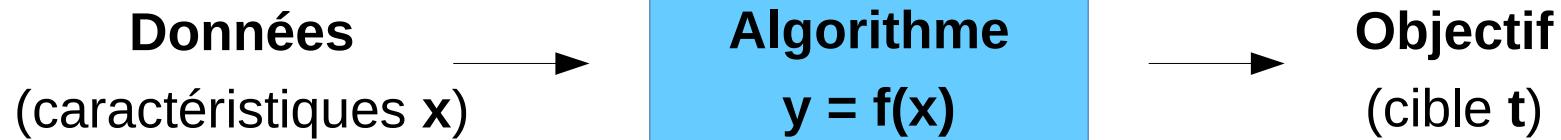


Classification

Dog

```
[[[ 7.4280e-02,  1.4022e-01, -2.2258e-02,  ..., -2.0172e-01,
    1.6240e-01,  5.5748e-02],
   [-1.1771e-02, -1.1327e-01,  3.0360e-01,  ...,  4.6299e-01,
    3.4765e-02,  2.2633e-02],
   [ 2.2252e-02,  2.1568e-01, -3.5726e-01,  ..., -7.4589e-02,
    7.0776e-02,  1.3573e-01],
   ...,
   [ 1.1035e-01, -2.4609e-01,  1.9962e-01,  ...,  2.4133e-01,
    -2.1069e-01,  1.9942e-01],
   [ 2.9337e-02,  2.4997e-01,  1.0341e-02,  ..., -3.1368e-01,
    -1.6878e-01, -1.4741e-02],
   [ 4.4006e-02,  5.1292e-02,  5.0462e-02,  ..., -8.1194e-02,
    1.6043e-01, -5.7106e-03]]],
```

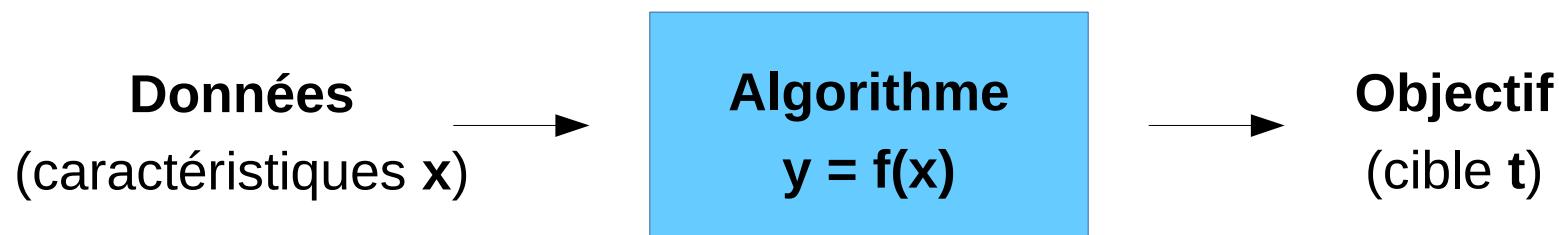
# Machine Learning



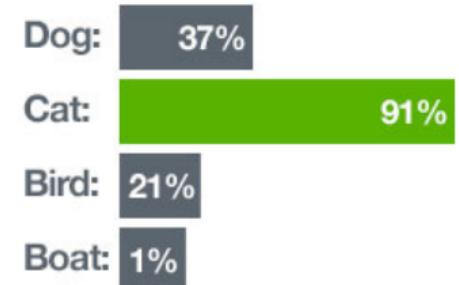
Entraînement

airplane  
automobile  
bird  
cat  
deer  
dog  
frog  
horse  
ship  
truck

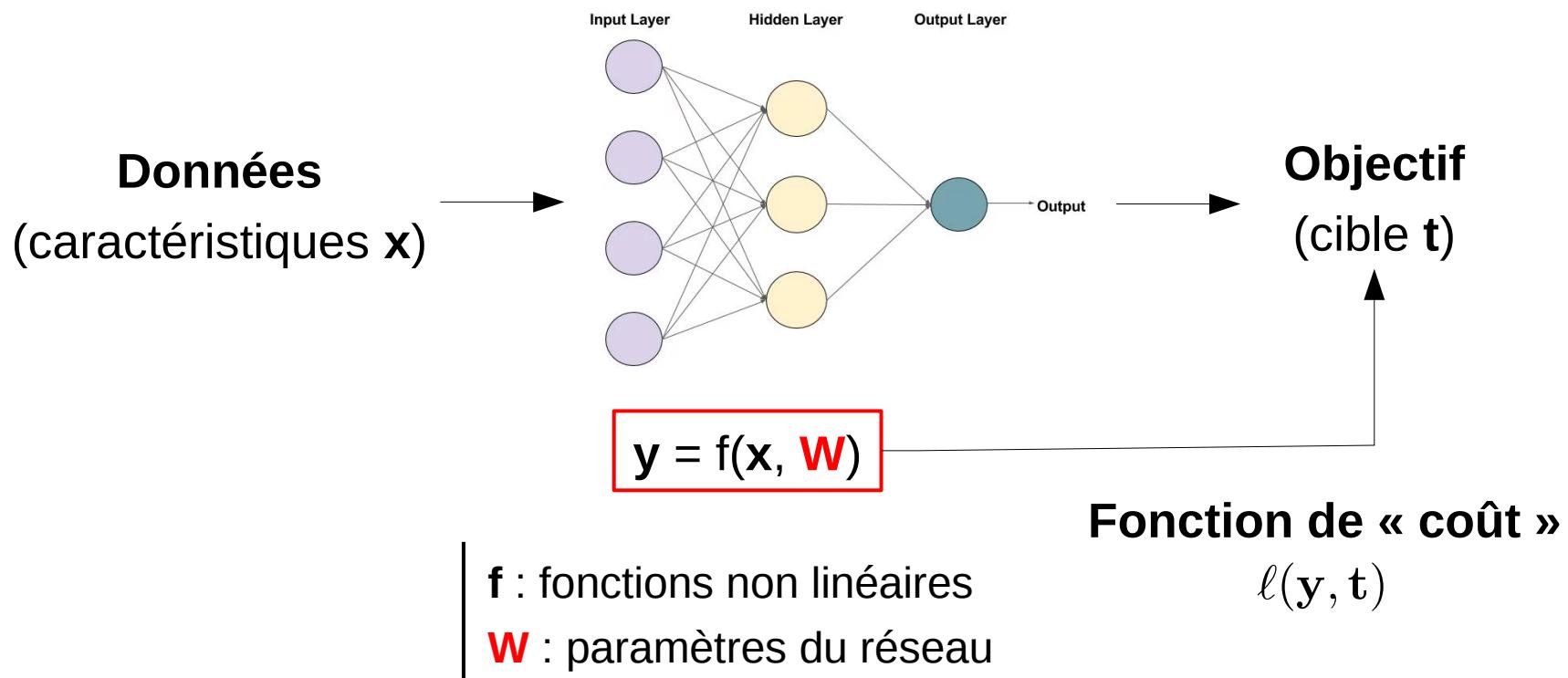
# Machine Learning



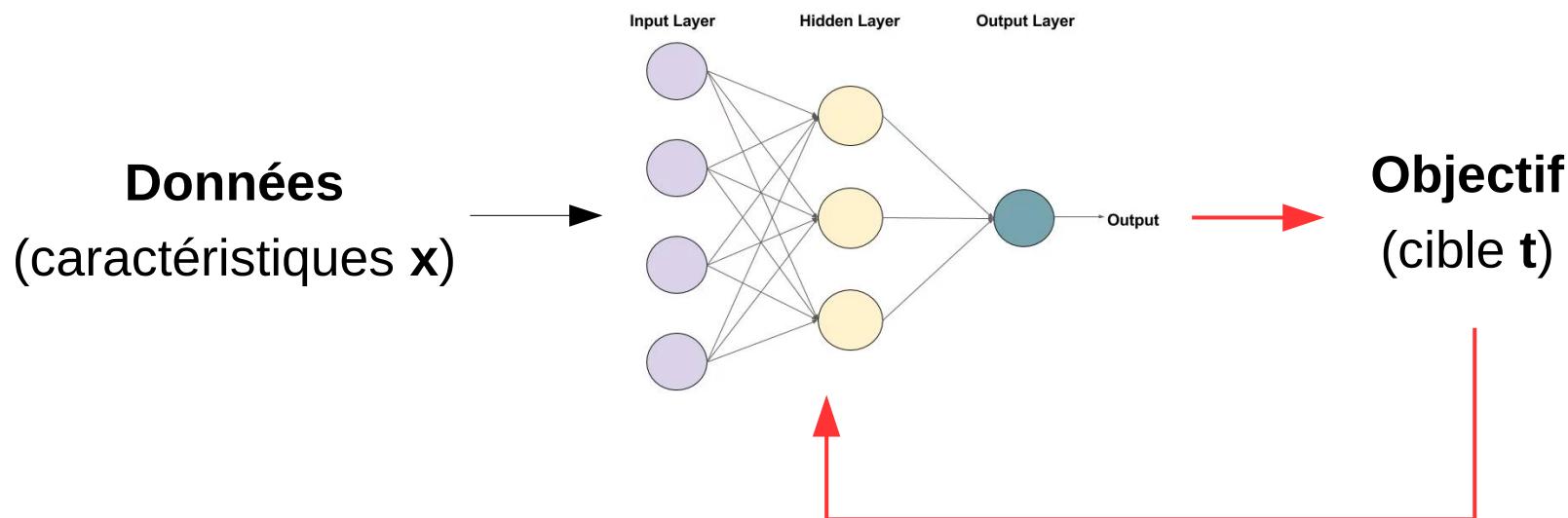
Test →



# Réseaux de neurones



# Réseaux de neurones

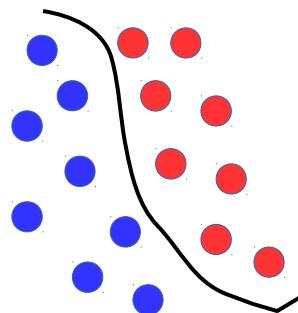


Mise à jour des poids **W**

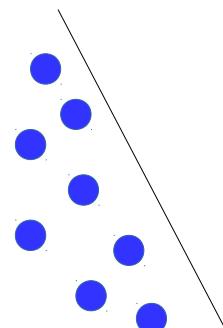
$$\mathbf{W} \rightarrow \mathbf{W} - \eta \sum_N \frac{\partial \ell(\mathbf{y}, \mathbf{t})}{\partial \mathbf{W}}$$

# Type d'apprentissage

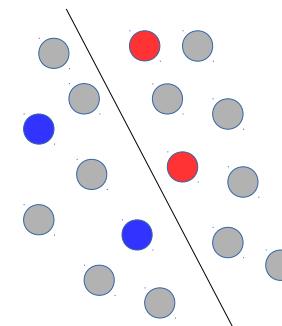
**Supervisé**  
**(classes connues)**



**Non supervisé**  
**(pas de classes)**



**Semi-supervisé**  
**(quelques classes)**

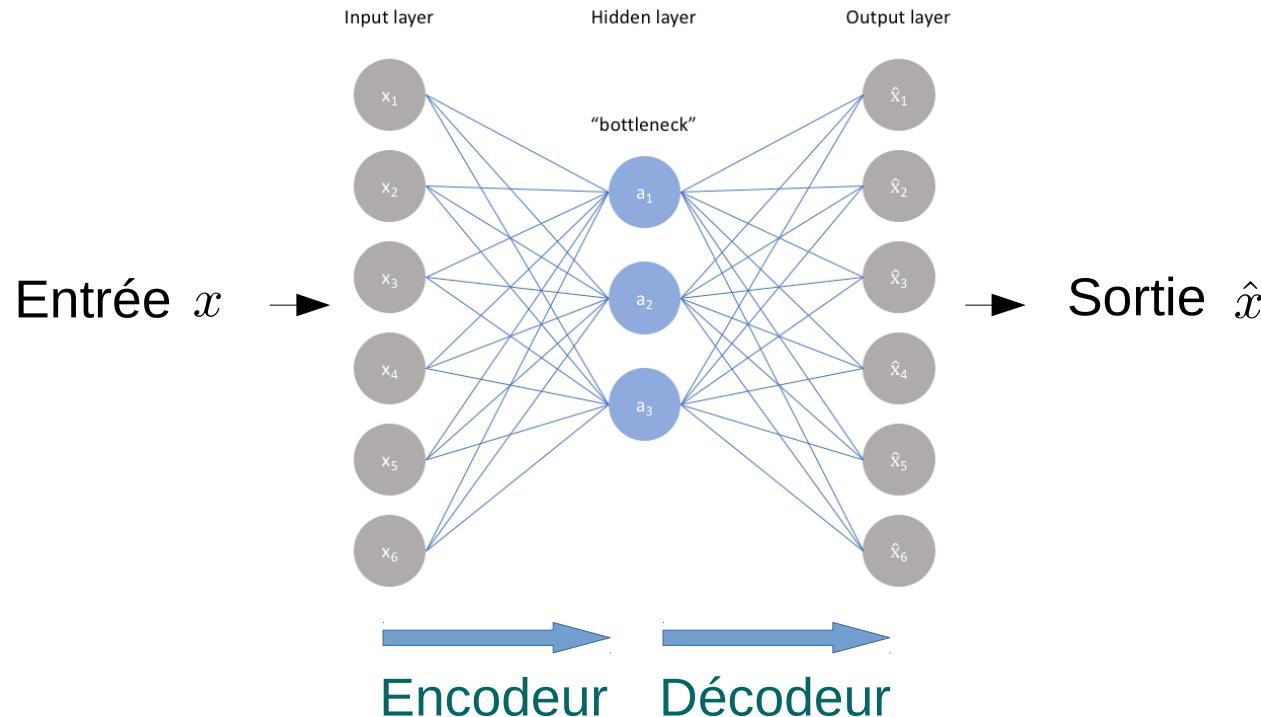


Détection d'anomalies

# Autoencodeurs

Un **autoencodeur** (AE) est entraîné pour **reproduire** les données d'entrées

$$x \rightarrow h = f(x) \rightarrow \hat{x} = g(h)$$



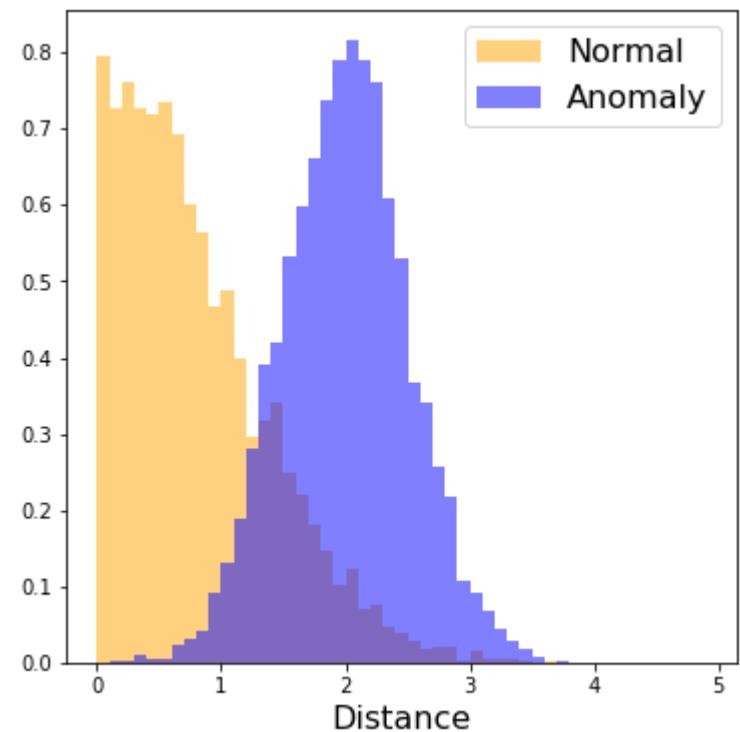
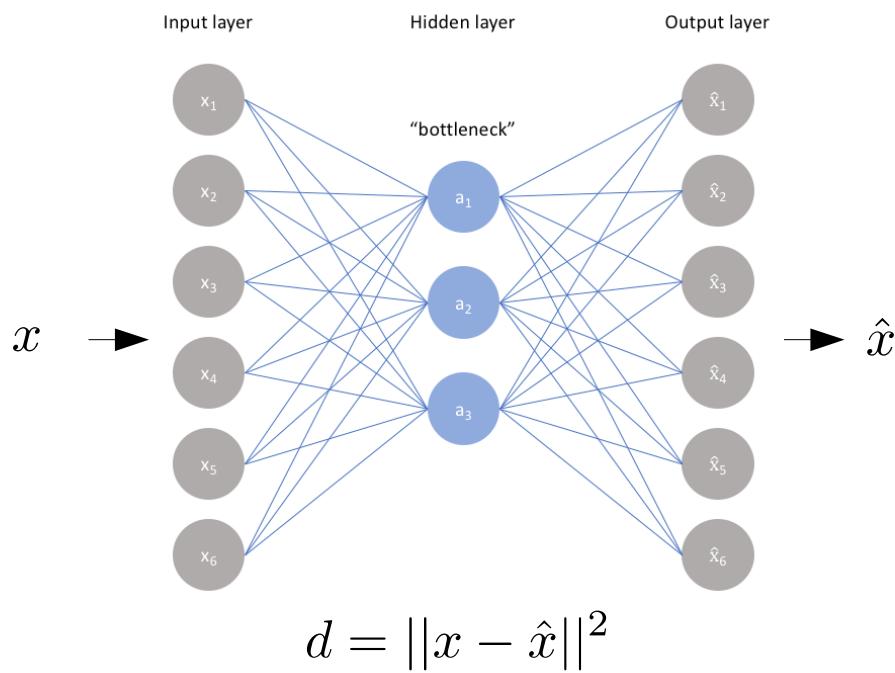
AE constraint afin d'apprendre les **composantes principales** des données

# Autoencodeurs

Un critère important est la qualité de la **reconstruction** :

⇒ **Distance** entre données d'entrée et de sortie :  $d = \|x - \hat{x}\|^2$

Un AE entraîné sur des données « **normales** » aura du mal à reconstruire une **anomalie** →  $d(\text{anomalie}) > d(\text{normale})$



# Exemple : détection de fraude à la CB



# Transaction de CB frauduleuses

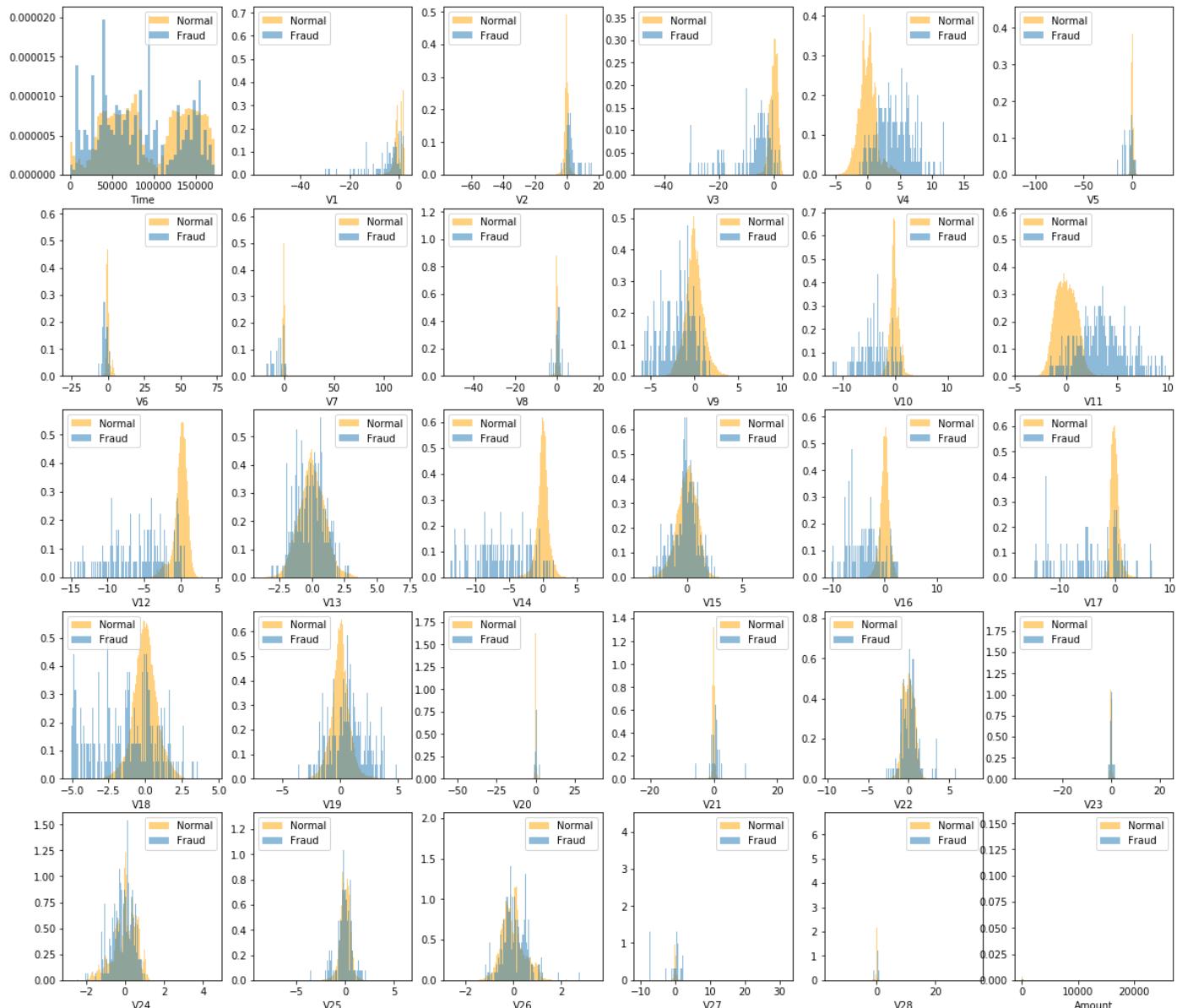
Données de transactions par carte sur une période de 2 jours (09/2013) :  
→ 492 fraudes sur 284 807 transactions.

Caractéristiques transformées (PCA) pour des raisons de confidentialité  
→  $x = 28$  caractéristiques + temps (en s) + montant (en €)

	Time	V1	V2	V3	V4	V5	V6	V7	\
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	
		V8	V9	...	V21	V22	V23	V24	\
0	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928		
1	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846		
2	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281		
3	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575		
4	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267		
5	0.2660314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427		
6	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055		
7	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709		
8	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592		
9	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050		
		V25	V26	V27	V28	Amount	Class		
0	0.128539	-0.189115	0.133558	-0.021053	149.62		0		
1	0.167170	0.125895	-0.008983	0.014724	2.69		0		
2	-0.327642	-0.139097	-0.055353	-0.059752	378.66		0		
3	0.647376	-0.221929	0.062723	0.061458	123.50		0		
4	-0.206010	0.502292	0.219422	0.215153	69.99		0		
5	-0.232794	0.105915	0.253844	0.081080	3.67		0		
6	0.750137	-0.257237	0.034507	0.005168	4.99		0		
7	-0.415267	-0.051634	-1.206921	-1.085339	40.80		0		
8	0.373205	-0.384157	0.011747	0.142404	93.20		0		
9	-0.069733	0.094199	0.246219	0.083076	3.68		0		

→ Catégorie (0 : normal / 1 : fraude)

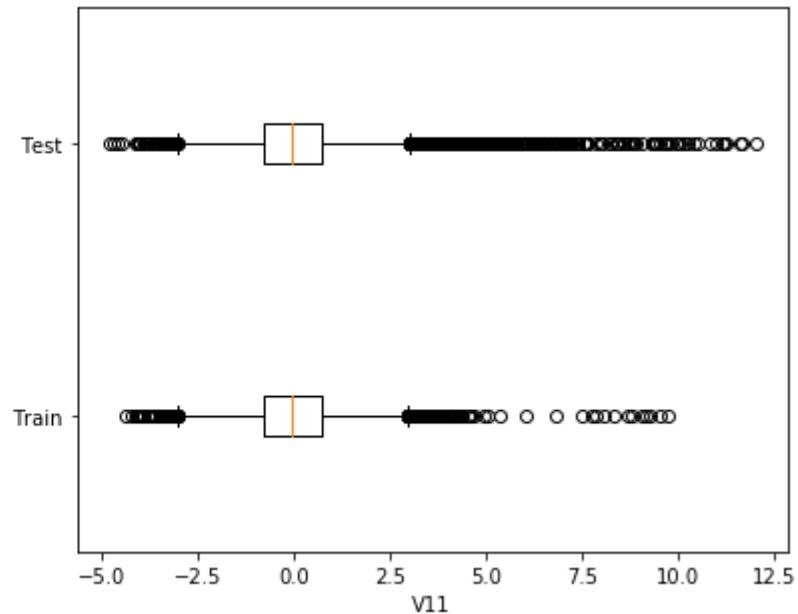
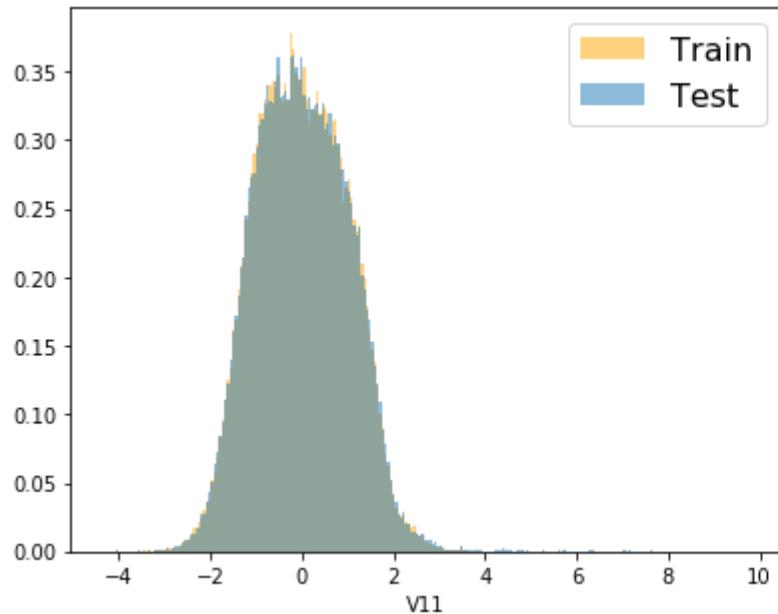
# Données (normales/fraudes)



# Échantillons train et test

Échantillon entraînement (« Train ») : 142k transaction **normales**

Échantillon test : 142k transaction **normales** + 492 fraudes (0,35%)



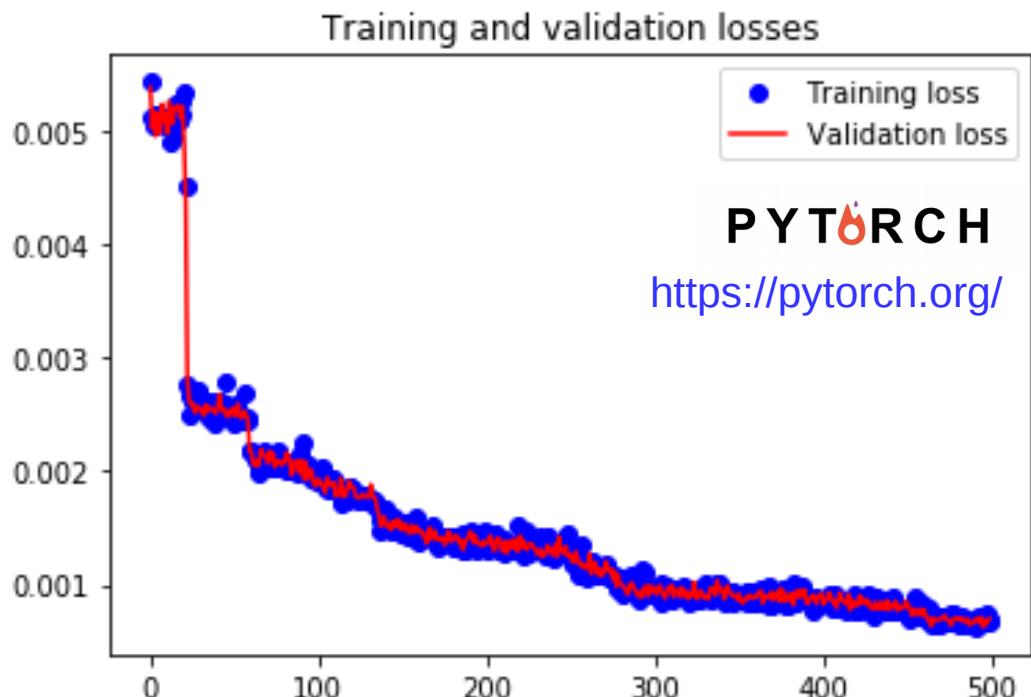
Difficile d'observer l'anomalie en regardant les caractéristiques individuelles !

# Entraînement de l'autoencodeur

L'autoencodeur est **entraîné** uniquement avec les données **normales** (Train)

## Hyperparamètres

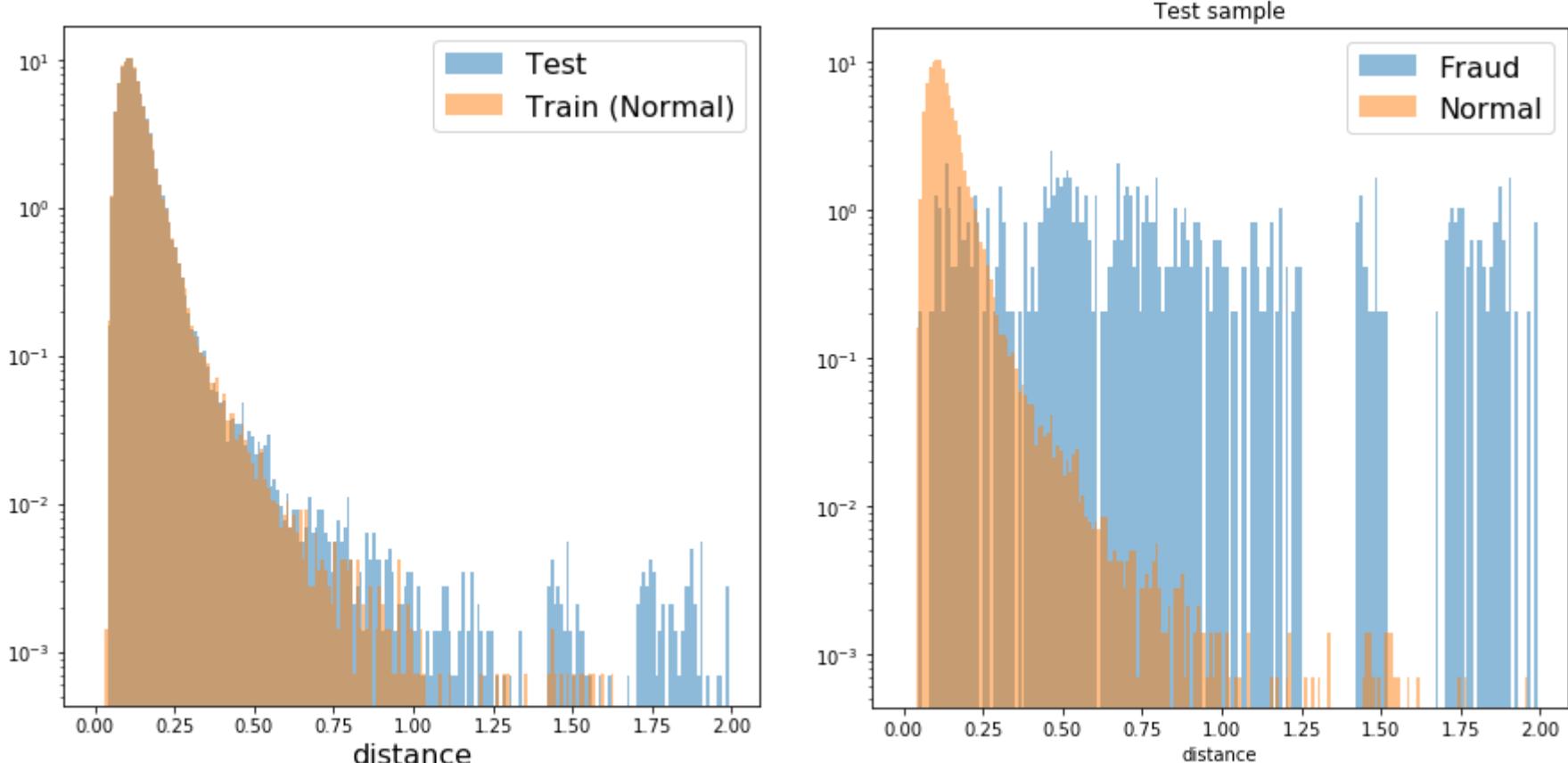
- Pre-processing : MinMAX
- num\_epochs = 500
- batch\_size = 2048
- hidden\_layer1 = 100
- hidden\_layer2 = 100
- encoding\_dim = 15



$$loss = \sum_{i \in N_{batch}} \|x_i - \hat{x}_i\|^2$$

# Distances après entraînement

## Distribution des distances sur les données Train, Test et Fraud

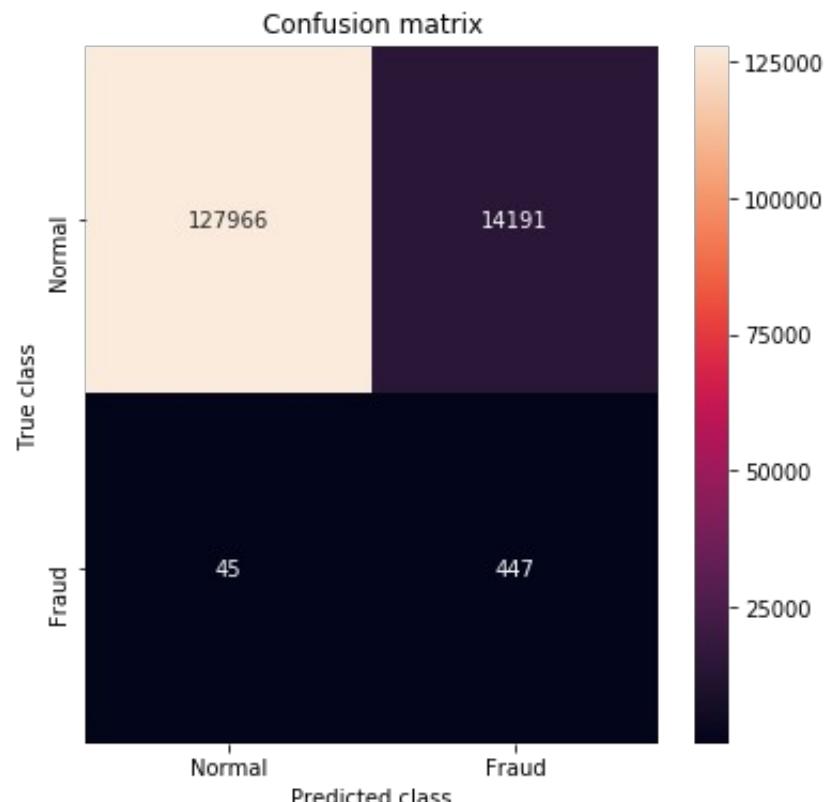
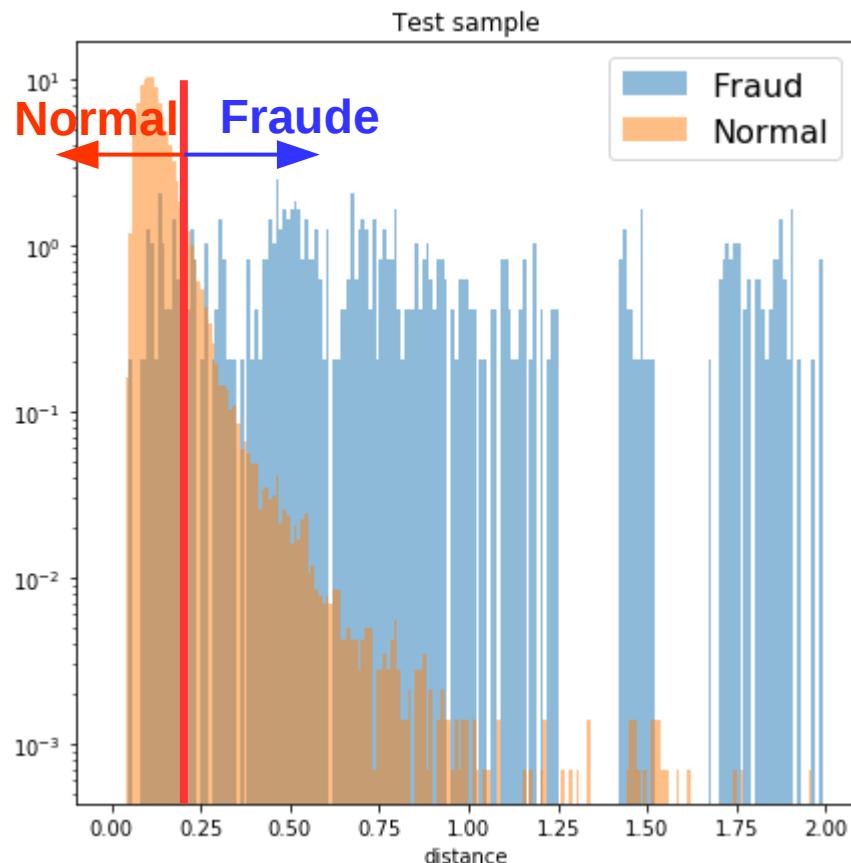


Train : 142k transaction normales

Test : 142k transaction normales + 492 fraudes

# Matrice de confusion

Qualité de la détection d'anomalies ? **Seuil** sur la distance calculée

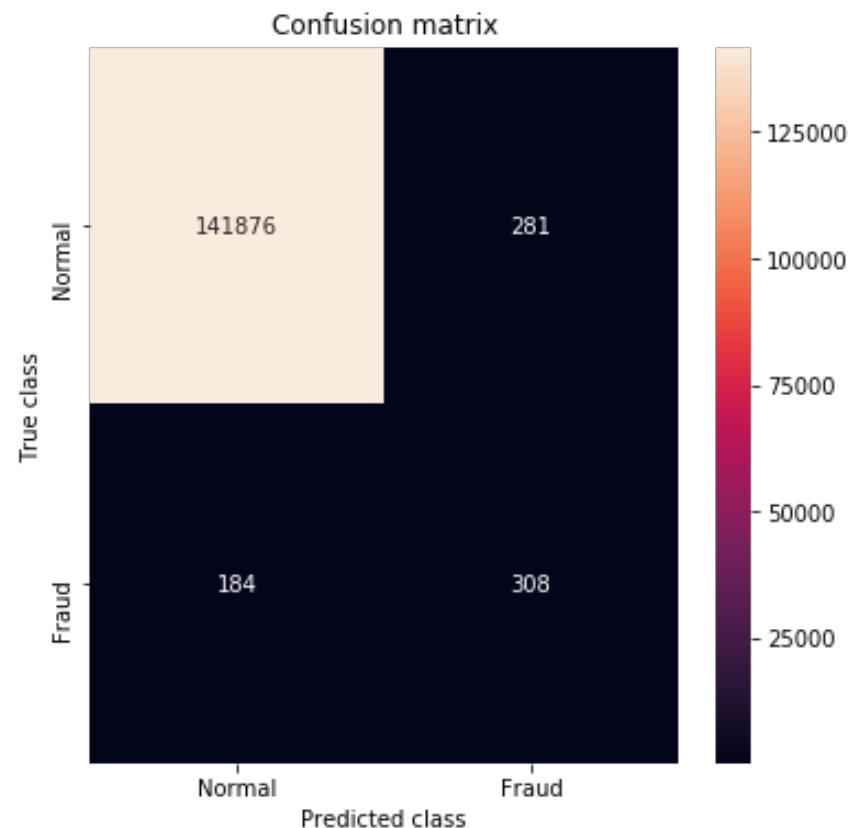
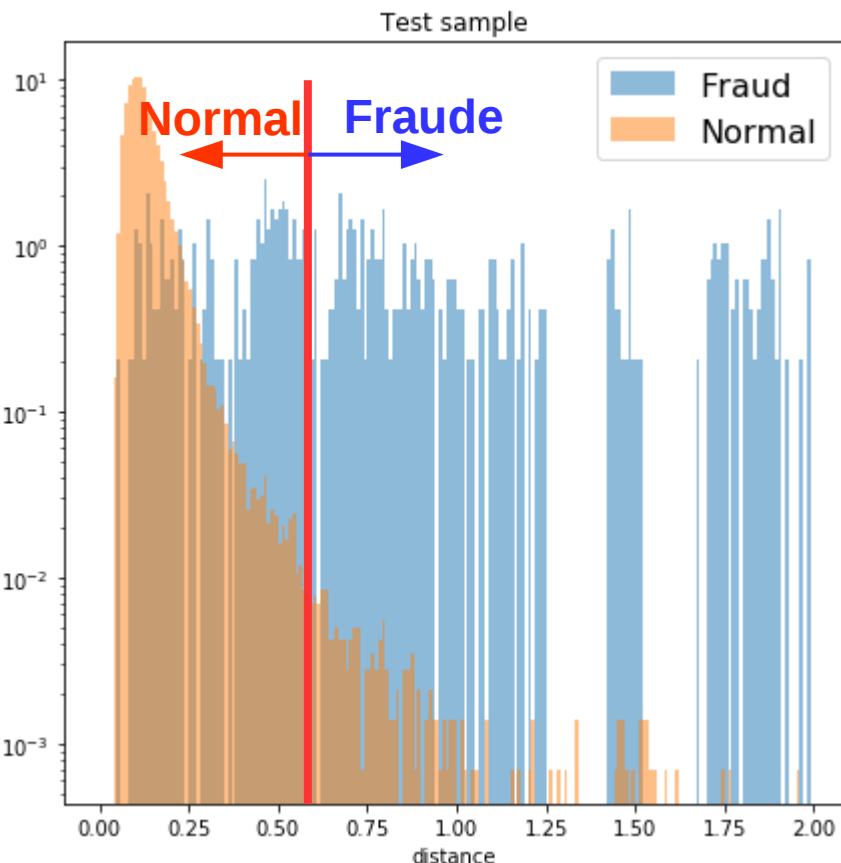


Pour un seuil :  $distance > 0.20$

- Taux de **faux positifs** (Normal → Fraude) = **10 %**
- Taux de **vrai positifs** (Fraude → Fraude) = **90.8 %**

# Matrice de confusion

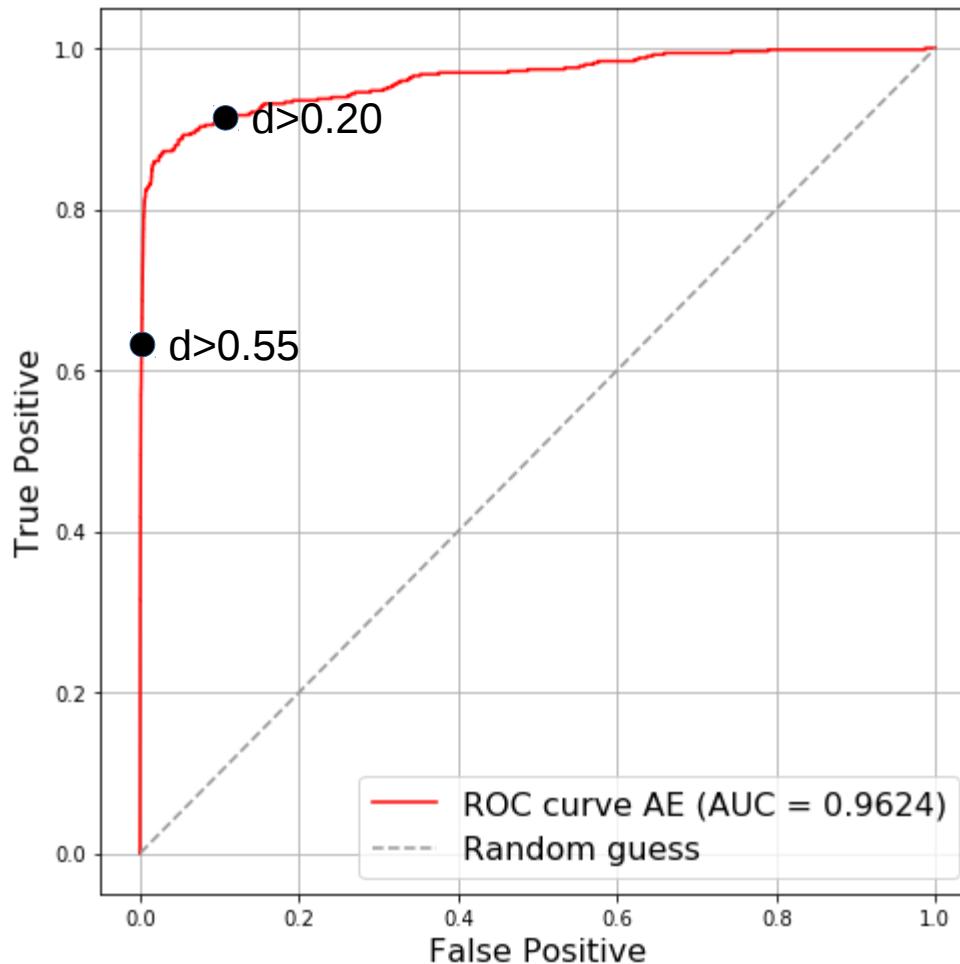
Qualité de la détection d'anomalies ? **Seuil** sur la distance calculée



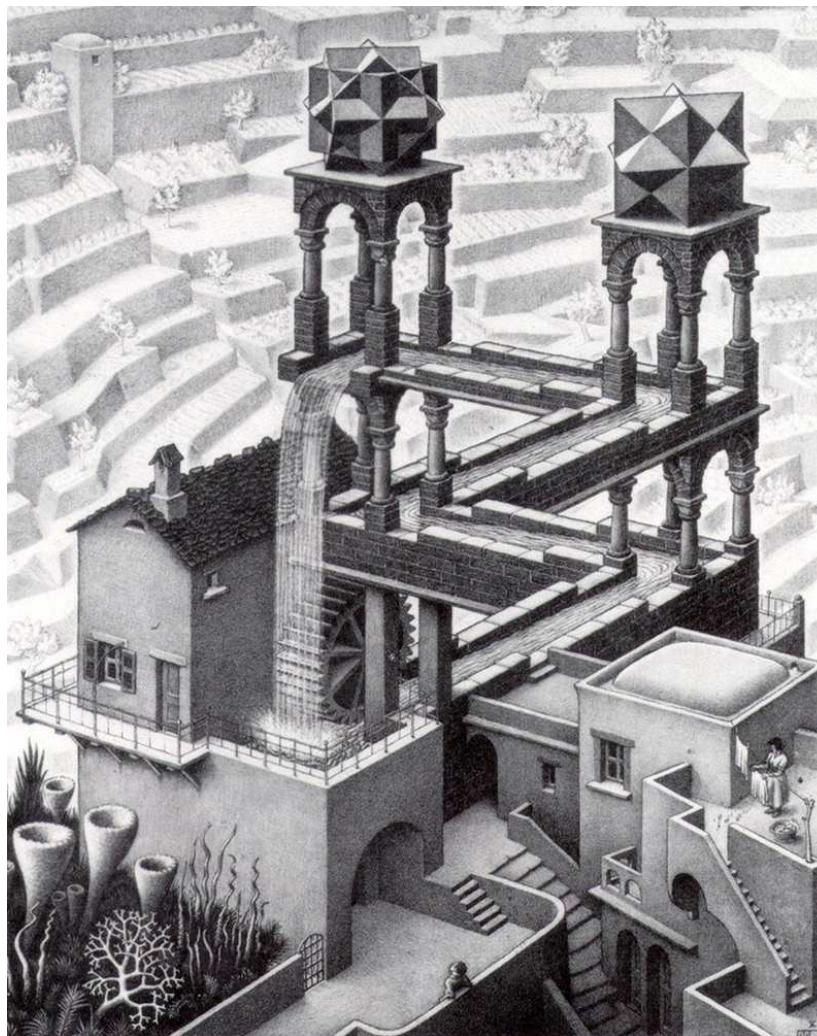
Pour un seuil :  $distance > 0.55$

- Taux de **faux positifs** (Normal → Fraude) = **0.20 %**
- Taux de **vrai positifs** (Fraude → Fraude) = **62.6 %**

## Qualité de la détection d'anomalies ? ROC et AUC

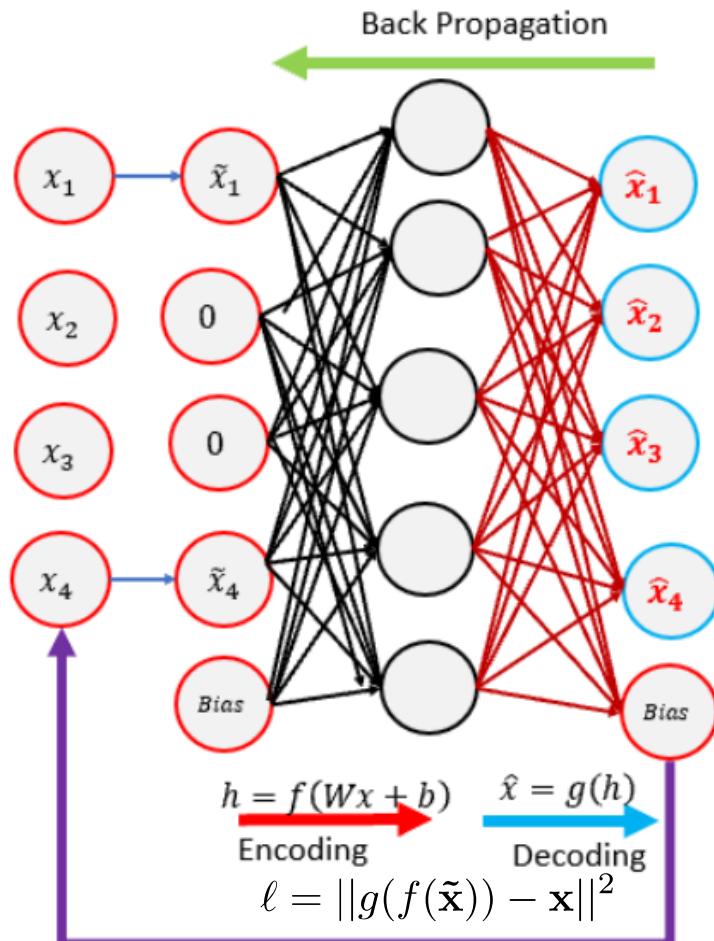


# Architectures AE avancées



# Autoencodeur débruiteur (DAE)

L'autoencodeur reçoit des données **corrompues** et est entraîné à **reproduire** les données **originales** non corrompues.

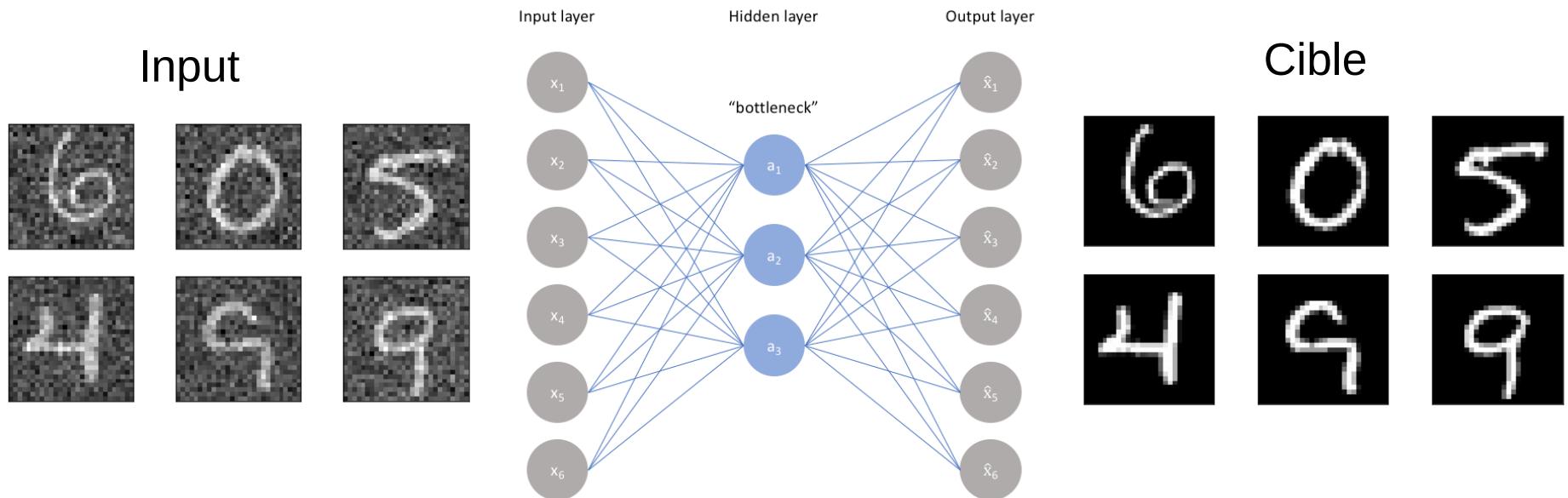


[image R. Khandelwal]

[goodfellow et al. <http://www.deeplearningbook.org>]

# Autoencodeur débruiteur (DAE)

Exemple de DAE entraîné pour corriger des images bruitées ([MNIST data](#)) :



Structure AE 784-350-350-350-784

num\_epochs = 20

batch\_size = 100

# Autoencodeur débruiteur (DAE)

Images reconstruites à partir de l'échantillon de test

Image originale

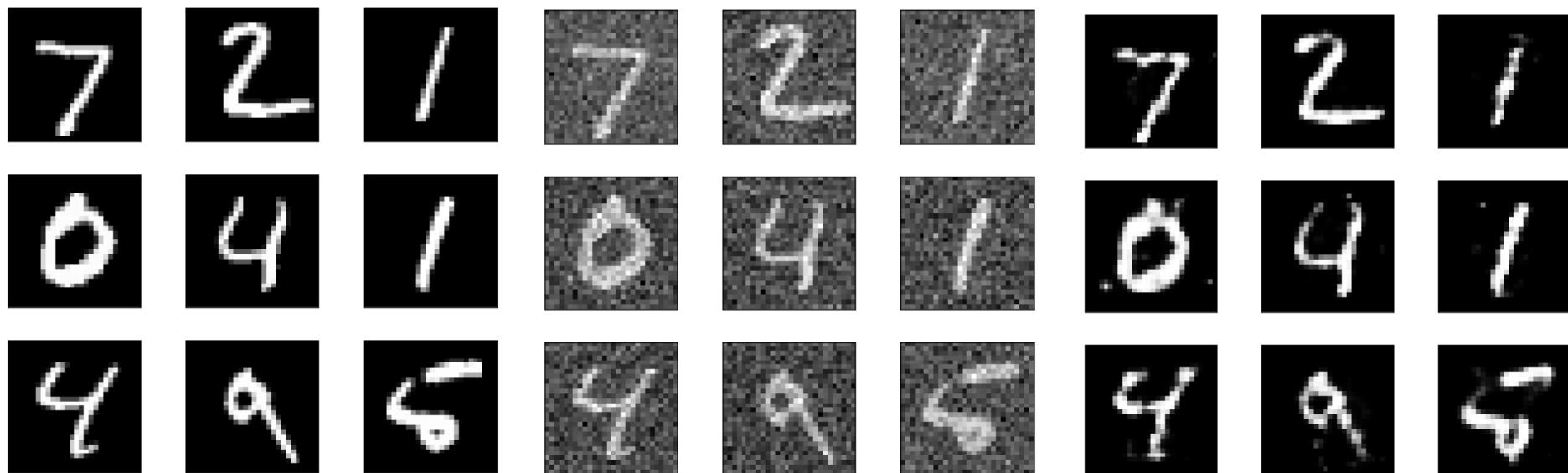
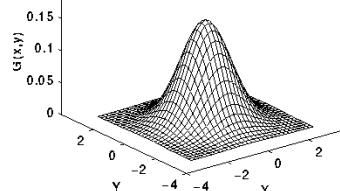
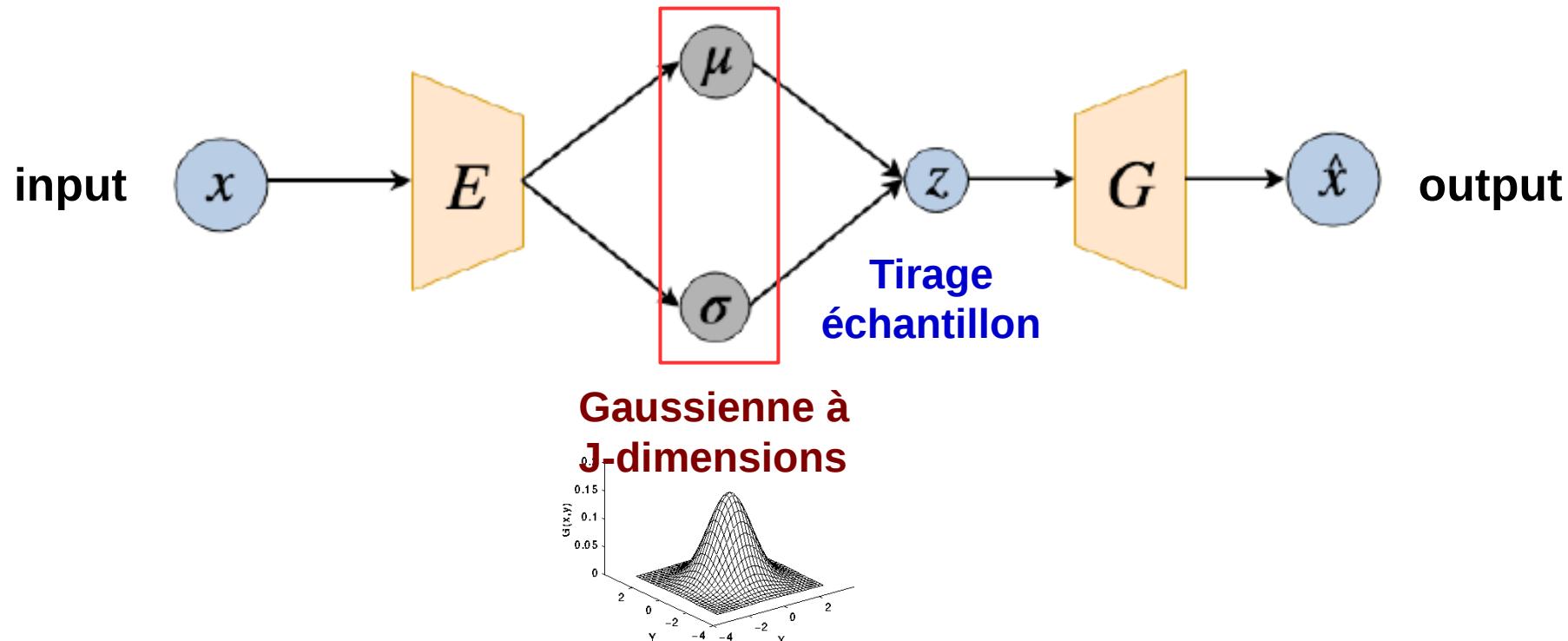


Image bruitée

Image corrigée

# Autoencodeur variationnel (VAE)

VAE [Kingma et al., 1312.6114] : modèles probabiliste utilisé pour la génération



**Fonction de coût :** Divergence Kullback-Leibler (différence entre le modèle dans l'espace de latence et une Loi Gaussienne centrée réduite)  
+ Erreur de reconstruction (différence input-output)

Pour plus d'info sur les VAE voir ces blogs: [ici](#), [ici](#) et [ici](#).

# Autoencodeur variationnel (VAE)

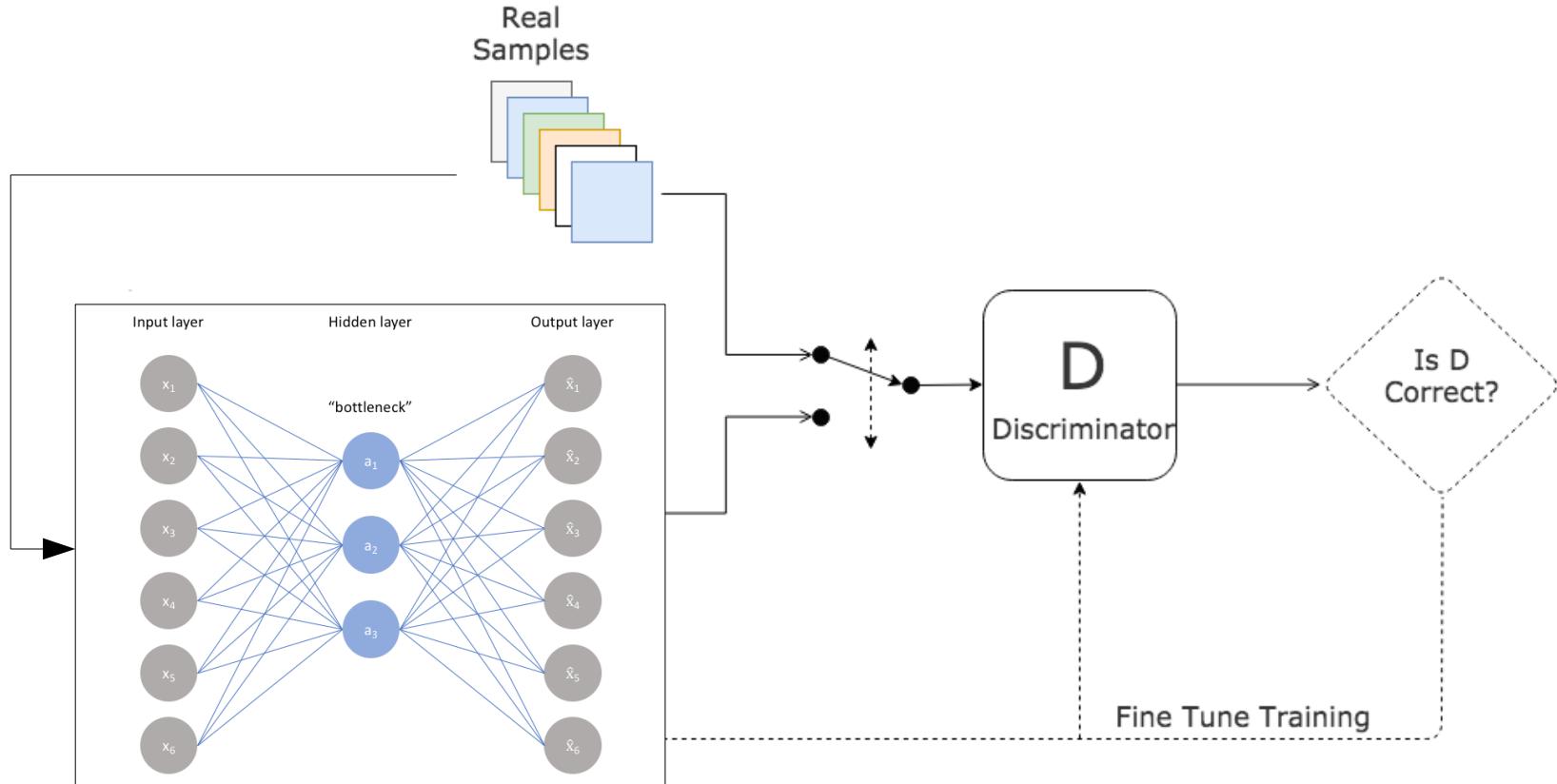
Exemple : génération d'images de chiffres manuscrits



Exemple tiré de ce [blog](#)

# Autoencodeur + réseau de neurone génératif antagoniste (GAN)

→ Stabilité et performances



Fonction de coût pour D : Binary crossentropy

Fonction de coût pour l'AE : Binary crossentropy +  $\epsilon \times$  erreur reconstruction

Autoencodeurs : structure spécifique de réseau de neurones

Plusieurs déclinaisons possibles : AE, DAE, GAN-AE, WAE ...

Applications pour de nombreux cas de figures

Exemples utilisés dans cette présentation:

<https://github.com/judonini/MLcourses2019/tree/master/tutorial>