

# **Auto-encodeurs et Apprentissage auto-supervisé**

Statistique en grande dimension et Apprentissage profond

---

Axel Carlier, **Juliette Chevallier**

2023 – 2024

INSA Toulouse, [juliette.chevallier@insa-toulouse.fr](mailto:juliette.chevallier@insa-toulouse.fr)

# Plan du cours

Introduction et Motivation

Auto-encodeurs : Premières définitions

Auto-encodeurs : Entrainement et Calibration

Auto-encodeurs : Exemples d'application

Apprentissage auto-supervisé

## Apprentissage supervisé

Dans le cadre de l'**apprentissage supervisé**, on dispose d'observations et de leurs étiquettes (appelées encore cibles (**target**), catégories ou **labels**) qui constituent un ensemble d'apprentissage. On note :

$$\mathcal{D} = \left\{ (\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)}) \right\} .$$

(i) *Classification,*

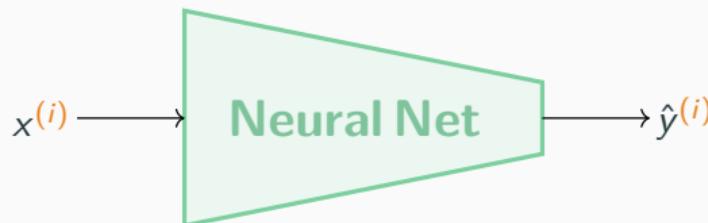
(ii) *Régression.*

## Apprentissage supervisé

Dans le cadre de l'**apprentissage supervisé**, on dispose d'observations et de leurs étiquettes (appelées encore cibles (**target**), catégories ou **labels**) qui constituent un ensemble d'apprentissage. On note :

$$\mathcal{D} = \left\{ (\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)}) \right\}.$$

Les labels permettent d'enseigner à l'algorithme à établir des **correspondances** entre les *observations* et les *labels*.



- (i) Classification,
- (ii) Régression.

## Apprentissage non-supervisé

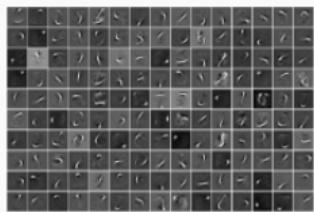
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

# Apprentissage non-supervisé

Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

**Des exemples :**

1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,



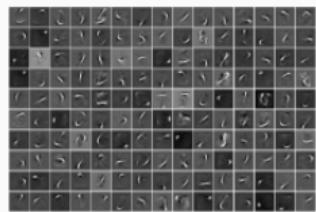
(1-2) Extraction de caractéristiques

# Apprentissage non-supervisé

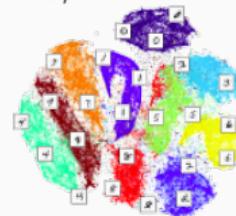
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

**Des exemples :**

1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. Réduction de dimension,



(1-2) Extraction de caractéristiques



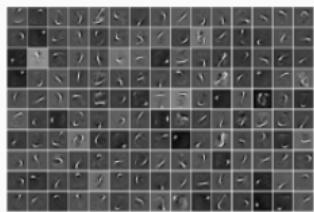
(3) Réduction de dimension

# Apprentissage non-supervisé

Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

## Des exemples :

1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. Réduction de dimension,
4. Compression de données,



(1-2) Extraction de caractéristiques



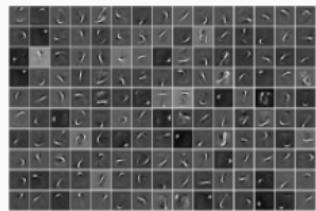
(3) Réduction de dimension

# Apprentissage non-supervisé

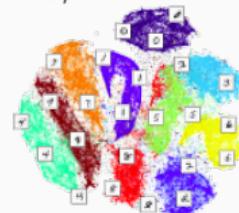
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

## Des exemples :

1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. Réduction de dimension,
4. Compression de données,
5. Détection de similitude dans des données,
6. Clustering (Partitionnement de données),



(1-2) Extraction de caractéristiques



(3) Réduction de dimension



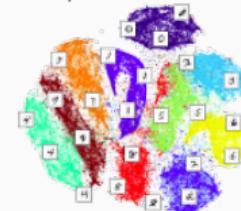
(6) Clustering

# Apprentissage non-supervisé

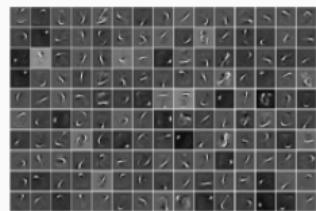
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

## Des exemples :

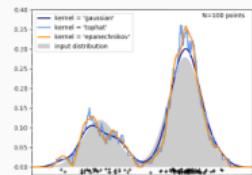
1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. Réduction de dimension,
4. Compression de données,
5. Détection de similitude dans des données,
6. Clustering (Partitionnement de données),
7. Éstimation de densité,



(3) Réduction de dimension



(1-2) Extraction de caractéristiques



(7) Éstimation de densité



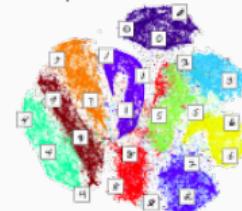
(6) Clustering

# Apprentissage non-supervisé

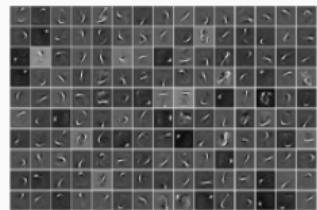
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

## Des exemples :

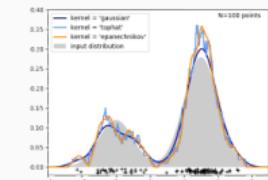
1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. Réduction de dimension,
4. Compression de données,
5. Détection de similitude dans des données,
6. Clustering (Partitionnement de données),
7. Éstimation de densité,
8. Génération de nouveaux exemples.



(3) Réduction de dimension



(1-2) Extraction de caractéristiques



(7) Éstimation de densité



(6) Clustering



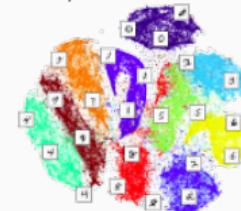
(8) Inpainting

# Apprentissage non-supervisé

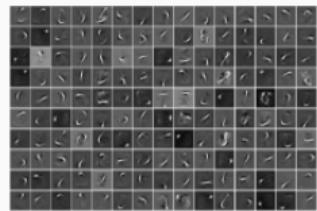
Dans le cadre de l'**apprentissage non supervisé**, on cherche à inférer de l'information à partir d'**observations uniquement** :  $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ .

## Des exemples :

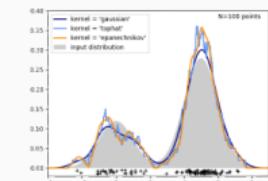
1. Trouver des structures cachées dans des données,
2. Extraction de caractéristique,
3. **Réduction de dimension**,
4. Compression de données,
5. Détection de similitude dans des données,
6. Clustering (Partitionnement de données),
7. Éstimation de densité,
8. Génération de nouveaux exemples.



(3) Réduction de dimension



(1-2) Extraction de caractéristiques



(7) Éstimation de densité



(6) Clustering

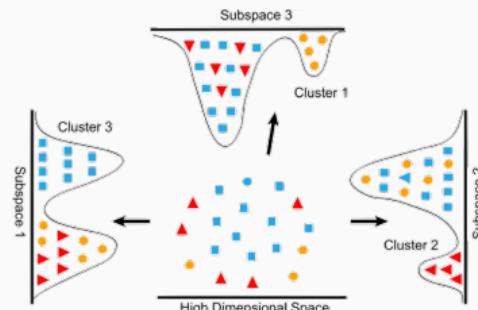


(8) Inpainting

# Réduction de dimension

**Idée** : Prendre des données dans un espace de grande dimension, et les remplacer par des données dans un espace de **plus petite dimension**.

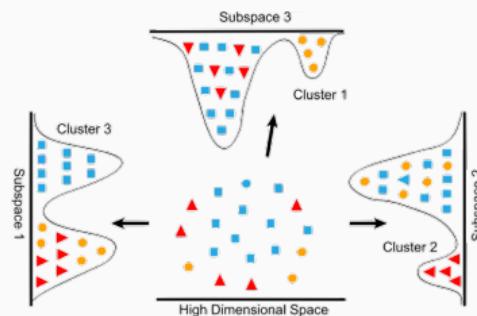
1. *Préserver l'information*, tout en réduisant le coût numérique,
2. *Améliorer la séparabilité* des différentes classes dans les données (clustering),
3. *Meilleure compréhension* des données par élimination de variables redondantes, etc.



# Réduction de dimension

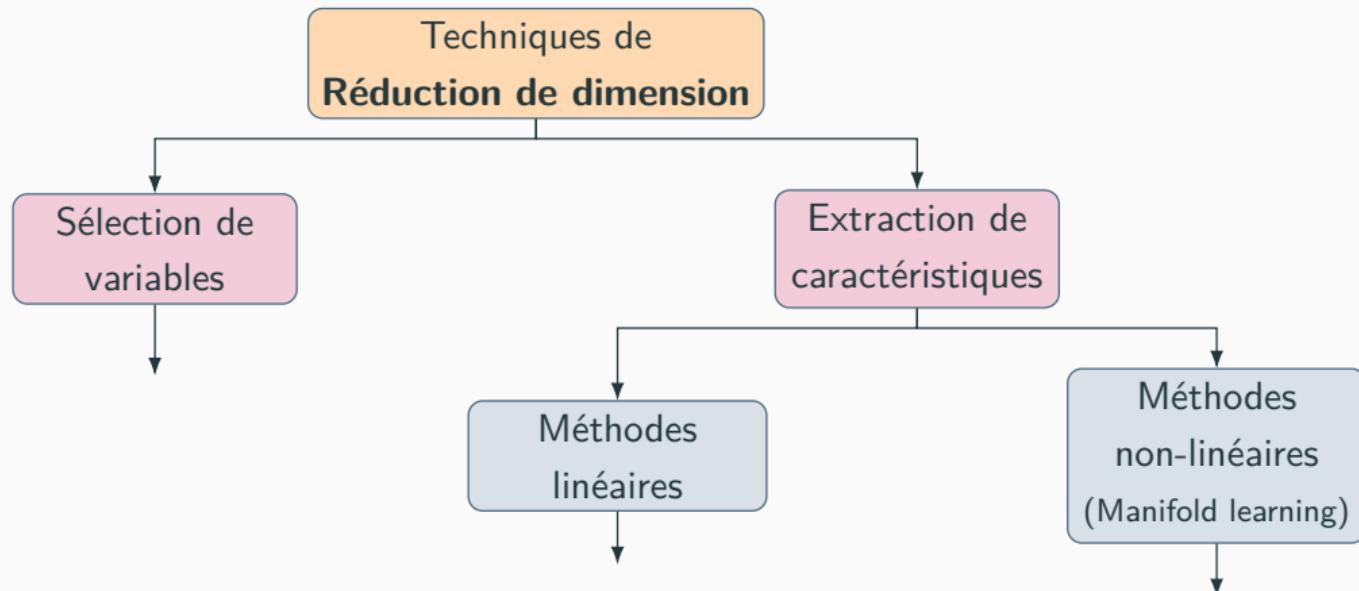
**Idée** : Prendre des données dans un espace de grande dimension, et les remplacer par des données dans un espace de **plus petite dimension**.

1. *Préserver l'information*, tout en réduisant le coût numérique,
2. *Améliorer la séparabilité* des différentes classes dans les données (clustering),
3. *Meilleure compréhension* des données par élimination de variables redondantes, etc.

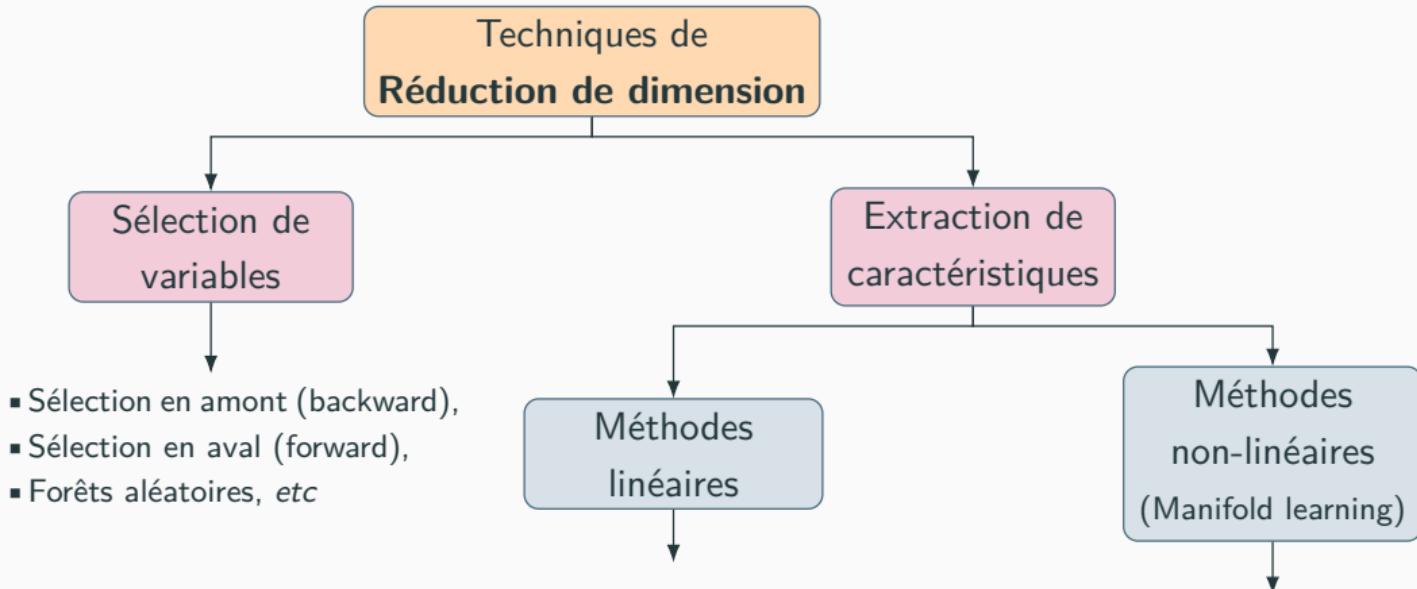


- **Sélection de caractéristiques** : Sélection d'un **sous-ensemble** de variables qui vont être conservées,
- **Extraction de caractéristiques** : Création de **nouvelles** variables plus pertinentes.

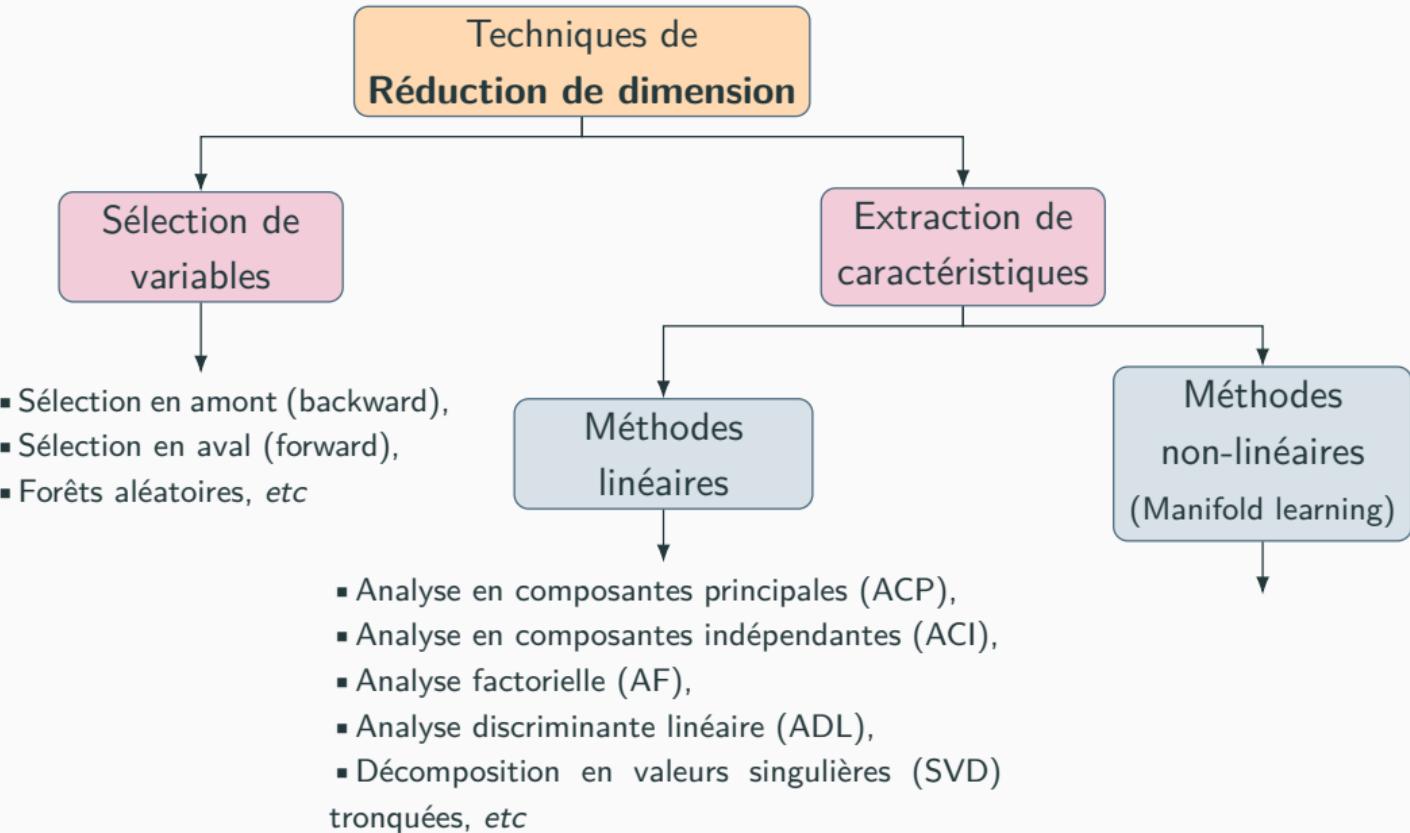
# Réduction de dimension



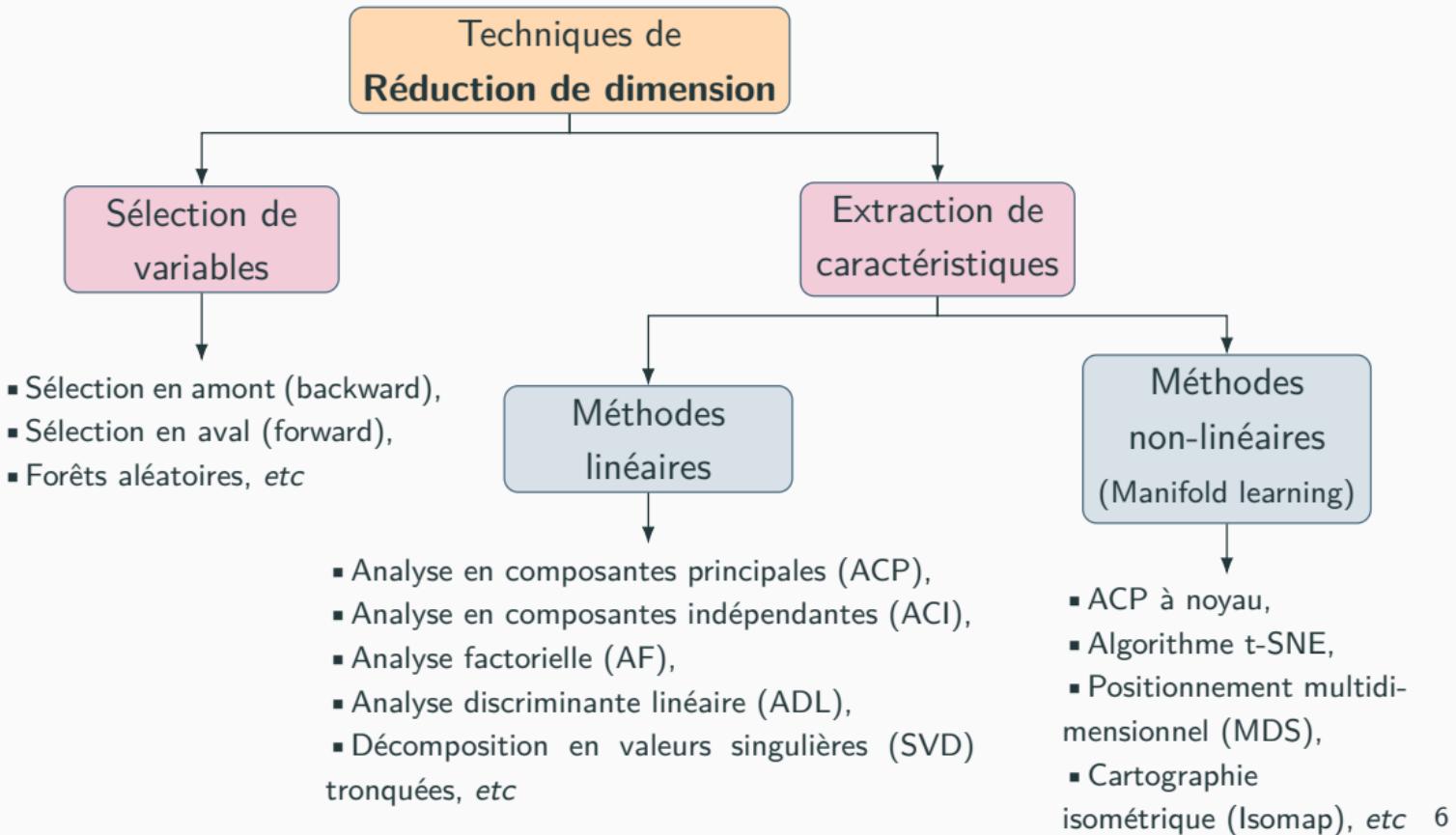
# Réduction de dimension



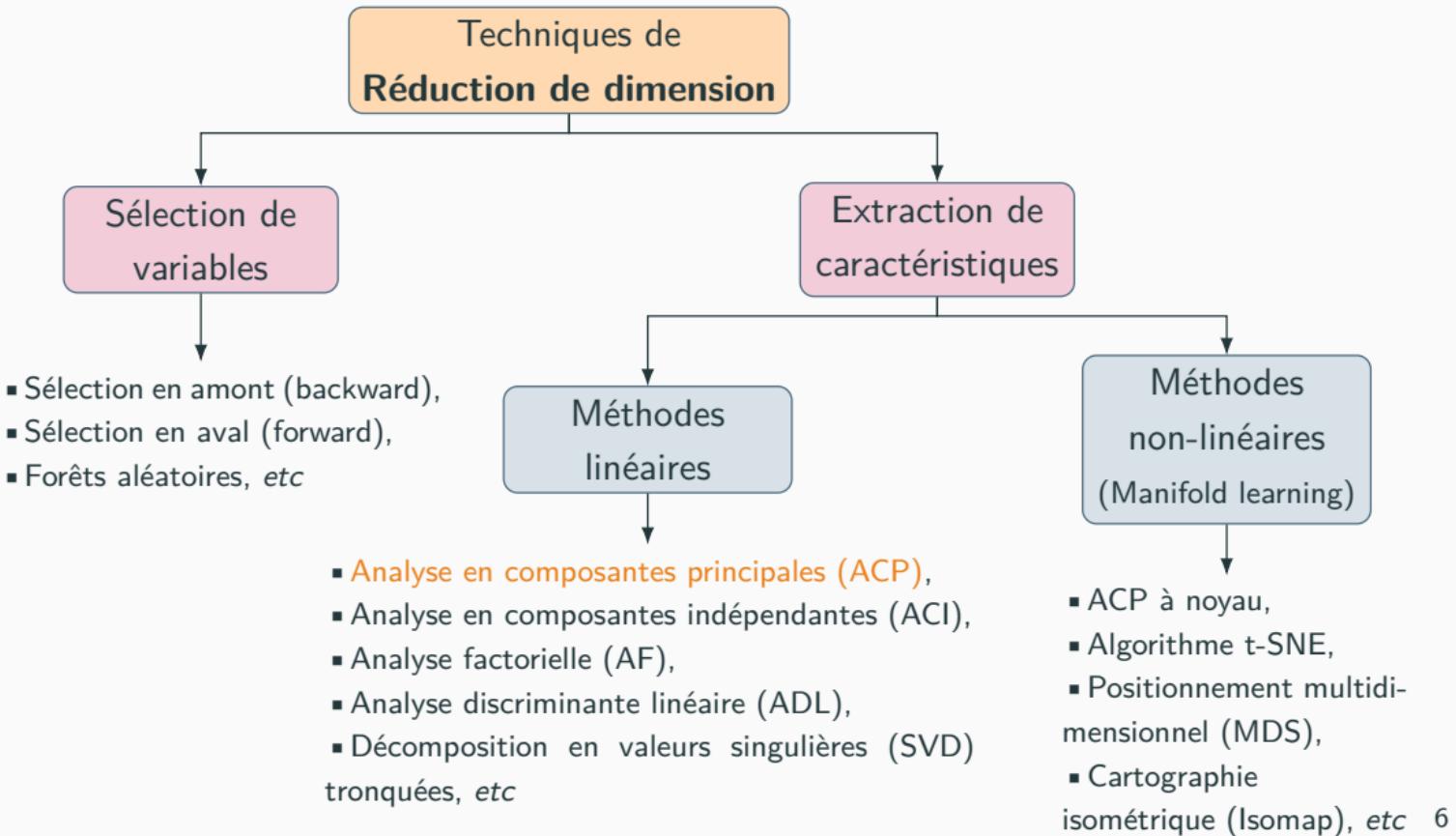
# Réduction de dimension



# Réduction de dimension



# Réduction de dimension

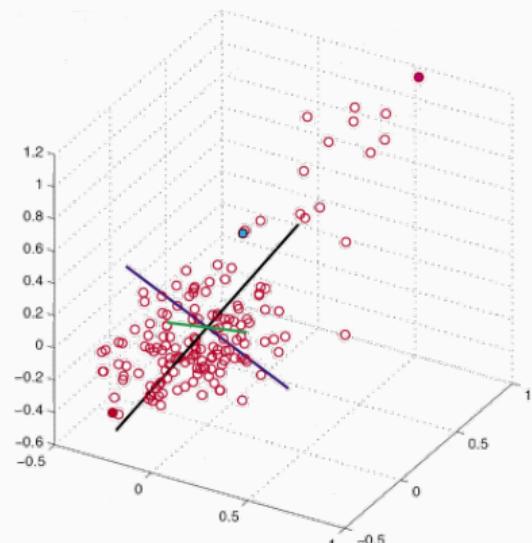
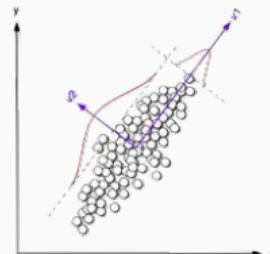


# Analyse en Composantes Principales (ACP)

**Idée** : Regrouper les variables de sorte à ce que la 1ère composante contienne un **maximum de variation**; la 2ème composante contient la deuxième plus grande quantité de variation, etc.

~~> *On peut alors conserver un petit nombre de composantes nouvellement créées, tout en conservant un maximum de variation.*

**En pratique** : On diagonalise la matrice de variance-covariance empirique de l'échantillon renormalisé.

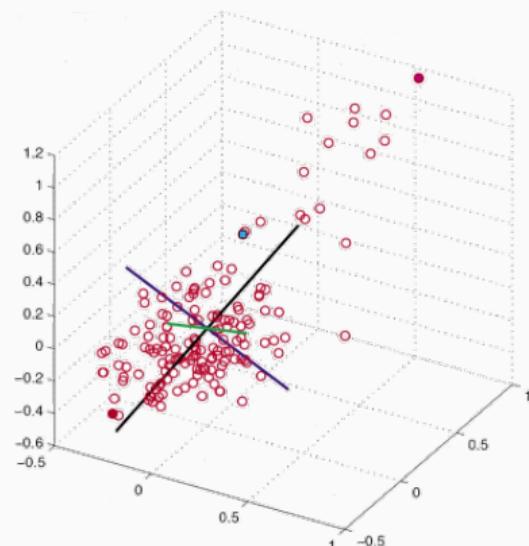
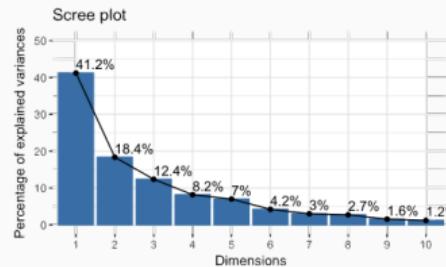
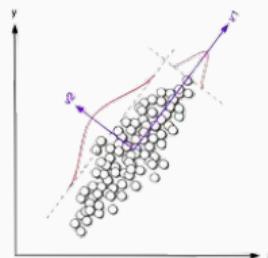


# Analyse en Composantes Principales (ACP)

**Idée** : Regrouper les variables de sorte à ce que la 1ère composante contienne un **maximum de variation**; la 2ème composante contient la deuxième plus grande quantité de variation, etc.

~~> *On peut alors conserver un petit nombre de composantes nouvellement créées, tout en conservant un maximum de variation.*

**En pratique** : On diagonalise la matrice de variance-covariance empirique de l'échantillon renormalisé.



# Plan du cours

Introduction et Motivation

Auto-encodeurs : Premières définitions

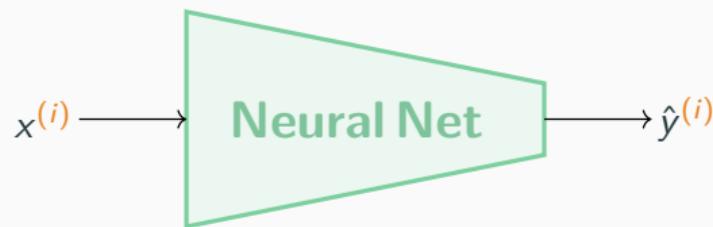
Auto-encodeurs : Entrainement et Calibration

Auto-encodeurs : Exemples d'application

Apprentissage auto-supervisé

## Comment apprendre sans label ?

En **apprentissage supervisé**, les **labels** permettent d'enseigner à l'algorithme à établir des **correspondances** entre les *observations* et les *labels*.



## Comment apprendre sans label ?

En **apprentissage supervisé**, les **labels** permettent d'enseigner à l'algorithme à établir des **correspondances** entre les *observations* et les *labels*.



**Question :** Comment apprendre sans annotations ?

## Comment apprendre sans label ?

En **apprentissage supervisé**, les **labels** permettent d'enseigner à l'algorithme à établir des **correspondances** entre les *observations* et les *labels*.

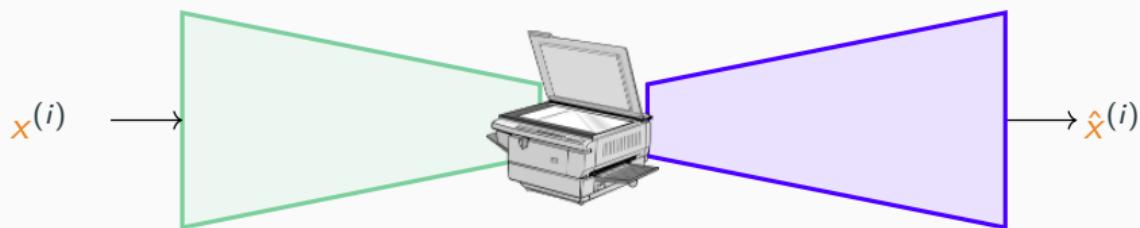


**Question** : Comment apprendre sans annotations ?

**Solution** : Chercher à prédire les données d'entrée !

## Comment apprendre sans label ?

En **apprentissage supervisé**, les **labels** permettent d'enseigner à l'algorithme à établir des **correspondances** entre les *observations* et les *labels*.



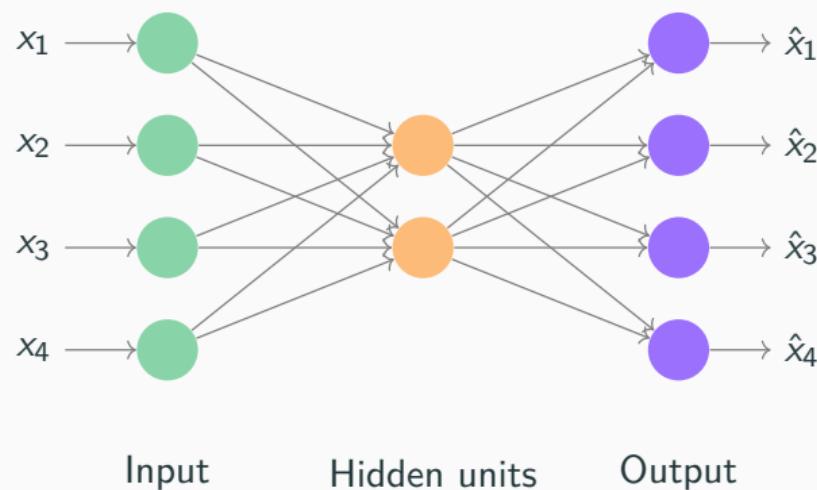
**Question** : Comment apprendre sans annotations ?

**Solution** : Chercher à prédire les données d'entrée !

Mais, avec une structure en **goulot d'étranglement**.

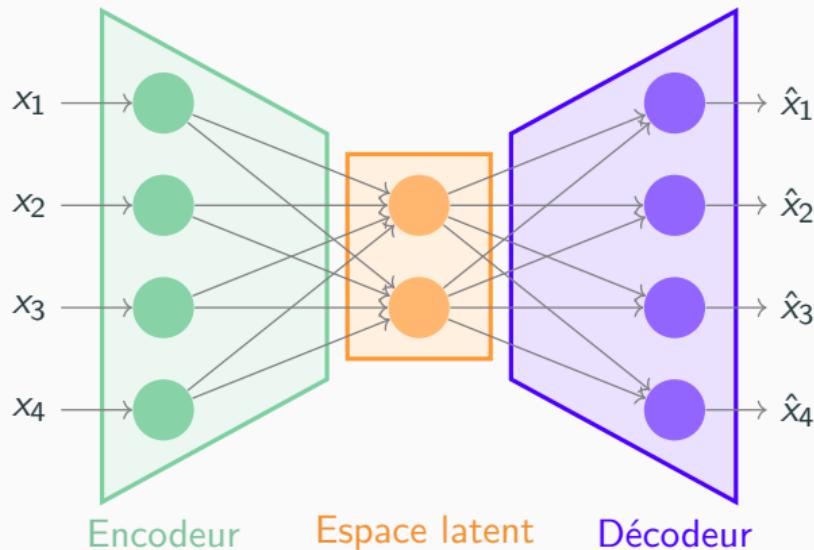
## Premier exemple : Auto-codeur entièrement connecté (MLP)

Perceptron multi-couches avec autant de neurones sur la couche de sortie que d'entrée, et peu de neurones cachés.



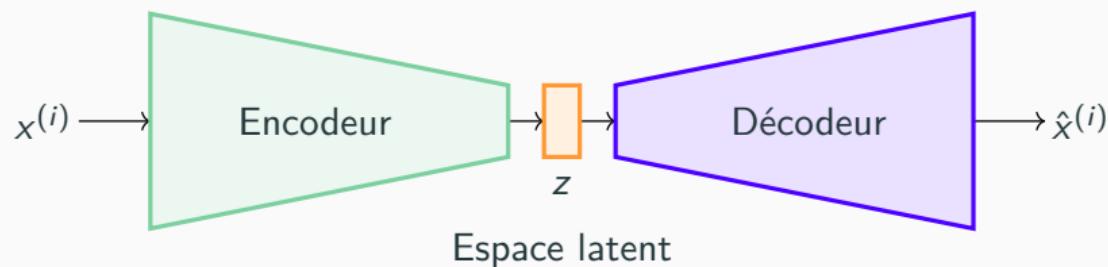
## Premier exemple : Auto-codeur entièrement connecté (MLP)

Perceptron multi-couches avec autant de neurones sur la couche de sortie que d'entrée, et peu de neurones cachés.

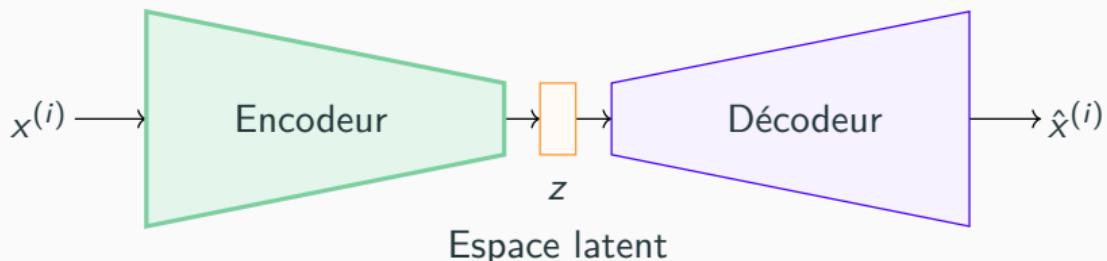


# Auto-encodeurs

Plus généralement :



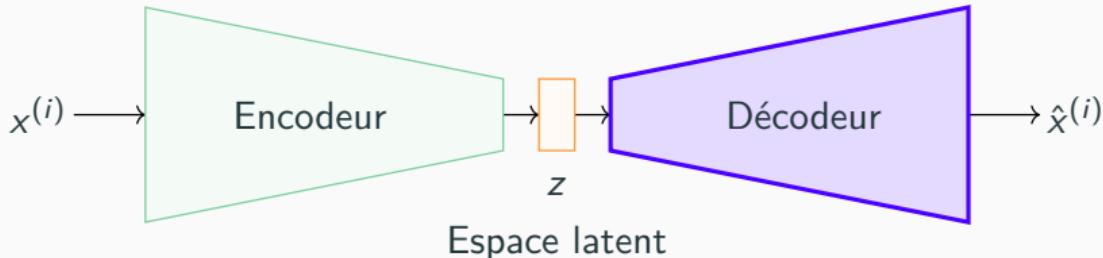
## Auto-encodeurs : L'encodeur



### *L'encodeur :*

- Extrait les **caractéristiques** principales (structure) de l'entrée ;
- **Comprime** le signal en réduisant la dimension ;
- Conserve suffisamment d'information discriminante pour permettre au décodeur de retrouver **l'information initiale** de manière convaincante.

## Auto-encodeurs : Le décodeur

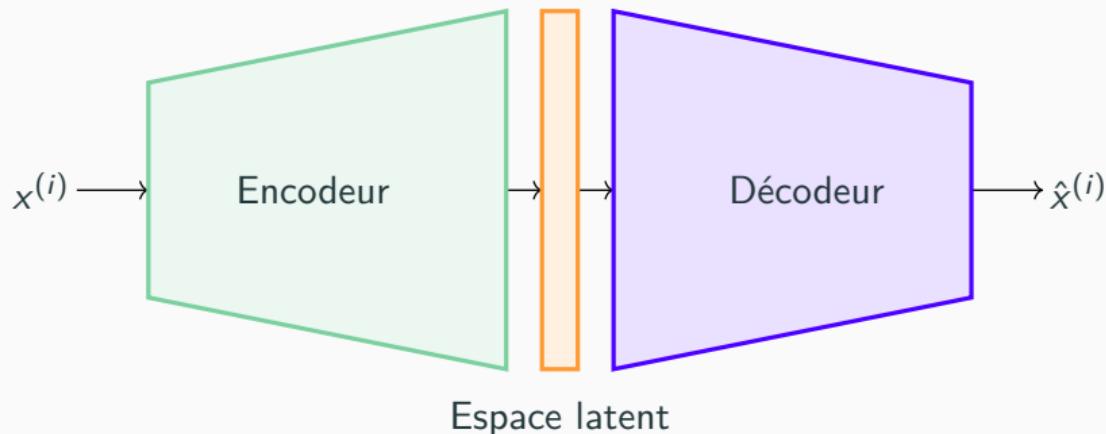


### Le décodeur :

- Reconstruit une approximation de l'entrée initiale à partir de sa *représentation latente* ;
- Est une forme de **modèle génératif** (cf. prochains cours).

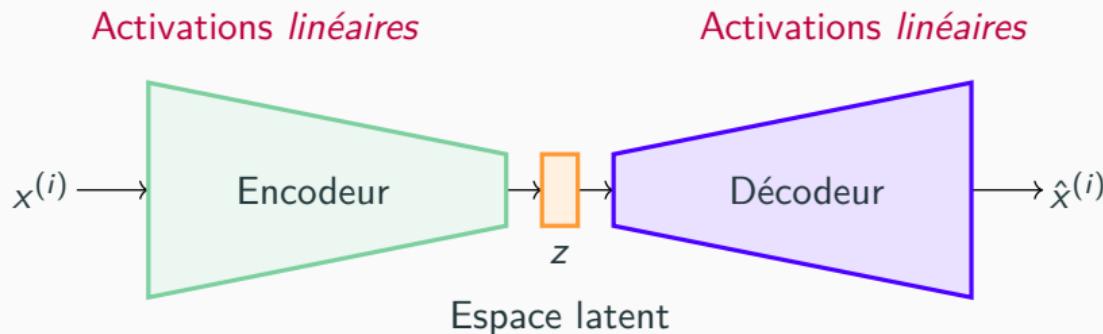
## Auto-encodeurs : Espace latent

Pourquoi un espace latent de petite dimension ? En fait, il n'y aurait pas vraiment de sens, ni d'intérêt, à construire des auto-encodeurs **surconditionnés**.



Dans ce cas, l'auto-encodeur n'aurait pas de difficulté à apprendre la fonction identité, ce qui n'a pas vraiment d'intérêt.

## Auto-encodeur pour la réduction de dimension

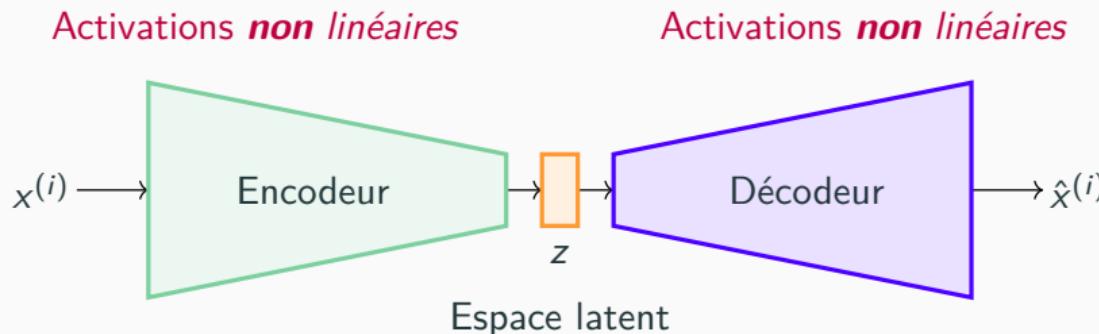


### ***Observation :***

Si toutes les fonctions d'activation des couches cachées de l'encodeur et du décodeur sont **linéaires**, l'espace latent appris tend vers l'**espace des composantes principales**.

~~> Dans ce cas, **l'auto-encodeur est équivalent à l'ACP !**

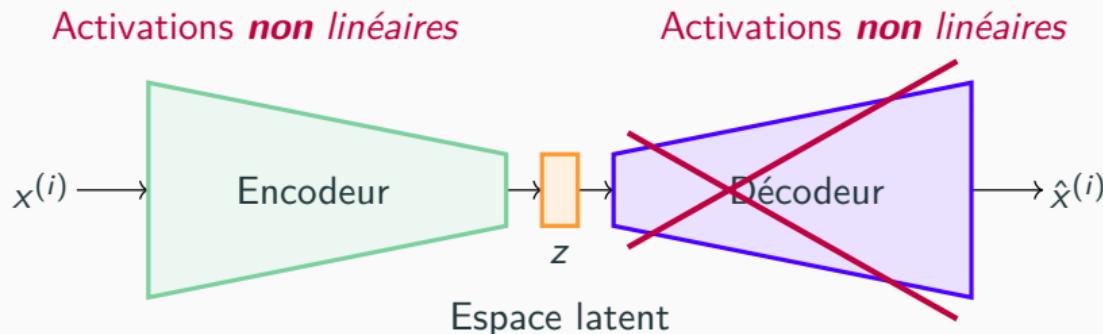
## Auto-encodeur pour la réduction de dimension



A contrario, si les fonctions d'activations de l'encodeur et du décodeur sont non linéaires, nous pouvons voir les auto-encodeurs comme une **extension de l'ACP** à des projections non-linéaires (t-SNE, MDS, etc.).

L'encodeur et le décodeur peuvent ainsi bénéficier de la grande capacité de représentation des réseaux de neurones pour projeter les données dans des espaces latents de plus faible dimension, et possédant de meilleures propriétés.

## Auto-encodeur pour la réduction de dimension

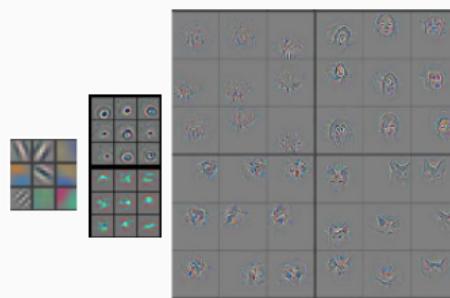


A contrario, si les fonctions d'activations de l'encodeur et du décodeur sont non linéaires, nous pouvons voir les auto-encodeurs comme une **extension de l'ACP** à des projections non-linéaires (t-SNE, MDS, etc.).

L'encodeur et le décodeur peuvent ainsi bénéficier de la grande capacité de représentation des réseaux de neurones pour projeter les données dans des espaces latents de plus faible dimension, et possédant de meilleures propriétés.

## Décodeur ←→ Extraction de caractéristiques

Nous en avons déjà vu des exemples d'**extraction de caractéristiques** !

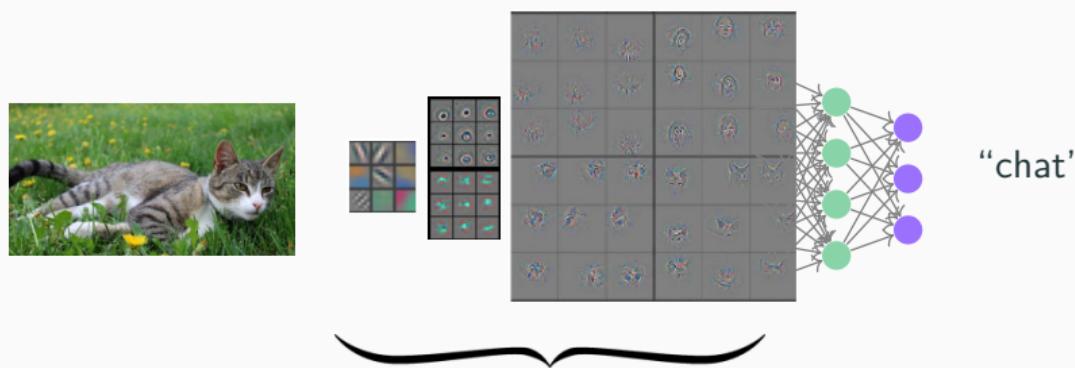


*Exemples : Filtres appris par le réseau AlexNet.*

## Décodeur $\longleftrightarrow$ Extraction de caractéristiques

Nous en avons déjà vu des exemples d'**extraction de caractéristiques** !

~ L'interprétation des filtres appris par les réseaux **convolutifs** sont à la base du **transfert d'apprentissage** :

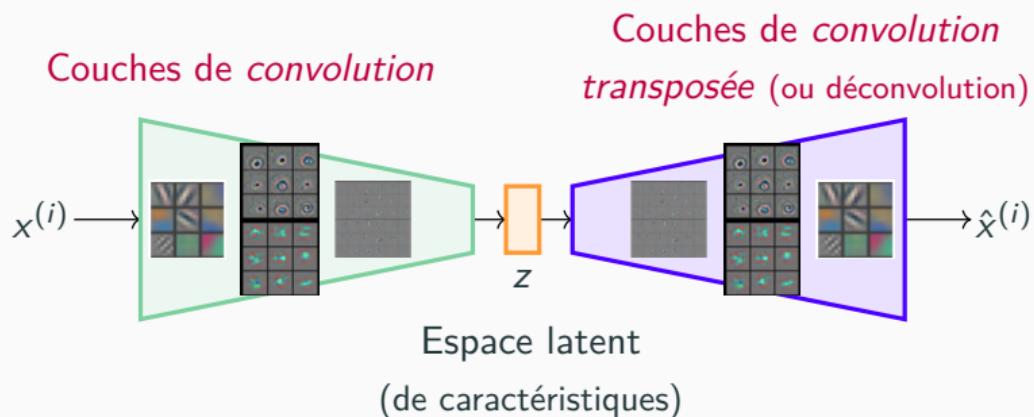


1- Gel des paramètres  
de l'extracteur de  
caractéristiques

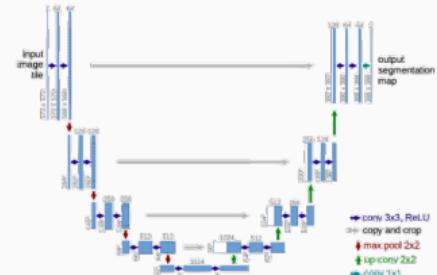
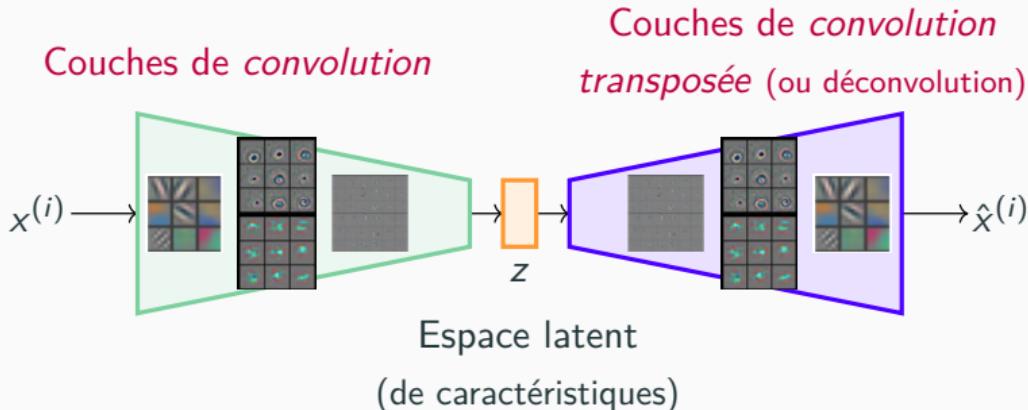
2- Entraînement des  
dernières couches  
du classifieur

Exemples : Filtres appris par le réseau AlexNet.

# Auto-encodeur convolutionnel



# Auto-encodeur convolutionnel



Le réseau U-Net peut, en altérant sa forme (omission des skip-connections), être vu comme un auto-encodeur convolutionnel.

# **Plan du cours**

Introduction et Motivation

Auto-encodeurs : Premières définitions

Auto-encodeurs : Entrainement et Calibration

Auto-encodeurs : Exemples d'application

Apprentissage auto-supervisé

## De la nécessité de *bien* choisir l'espace latent

**Attention !** Il y a un équilibre à trouver entre la **capacité** de représentation de l'encodeur et du décodeur et la **dimension** de l'espace latent !

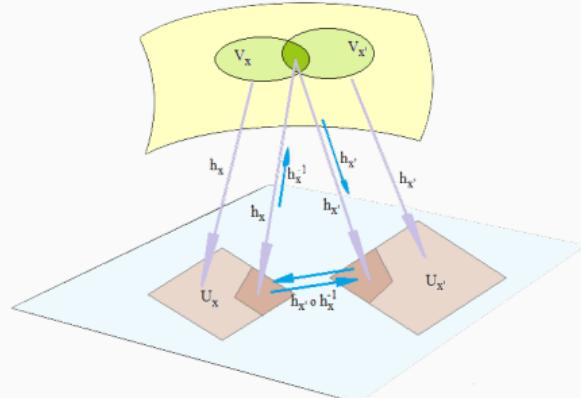
Avec un encodeur et un décodeur de très grande capacité, et un espace latent de dimension 1, on pourrait imaginer un cas dégénéré où l'encodeur apprendrait à associer à chaque donnée  $x^{(i)}$  son indice  $i$ .

Un tel auto-encodeur n'aurait aucune capacité de *généralisation*, et serait donc parfaitement inutile.

# Variétés (différentielles)

Les variétés sont des espaces généralisant les espaces vectoriels, en s'affranchissant du caractère linéaire de ceux-ci :

- des **courbes** en dimension 1,
- des **surfaces** en dimension 2,
- (hyper)-volumes en dimensions supérieures.



La plupart du temps, les données que nous traitons en apprentissage machine sont dans des espaces de très haute dimension, mais concentrées autour de variétés (exemple ci-dessous).

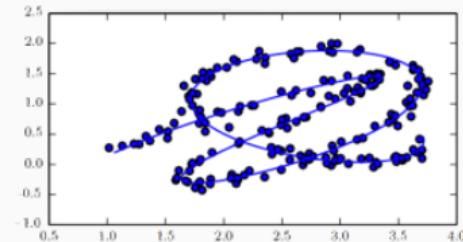
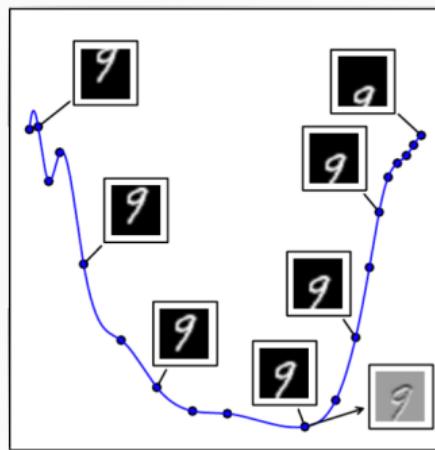


Image de [Goodfellow et al.] Deep Learning

## Variétés et Auto-encodeurs

**But** : Décrire, dans l'espace latent, la variété sur laquelle vivent nos données,  
*i.e.* trouver un **sous-espace** sur lequel on a une haute probabilité de rencontrer des données d'apprentissage.



## Variétés et Auto-encodeurs

**But** : Décrire, dans l'espace latent, la variété sur laquelle vivent nos données,  
i.e. trouver un **sous-espace** sur lequel on a une haute probabilité de rencontrer des données d'apprentissage.

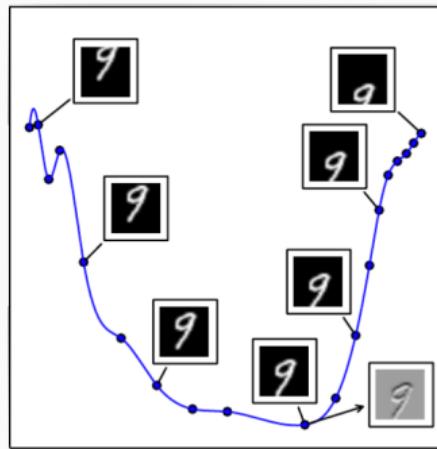


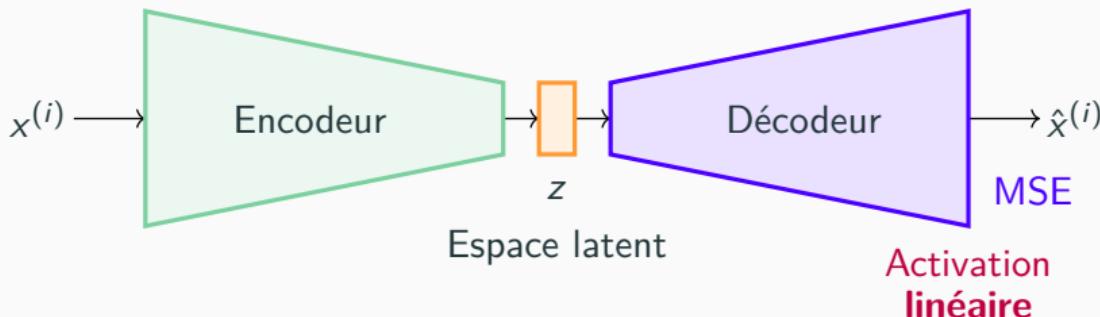
Image de [Goodfellow et al.] Deep Learning

**Exemple** : Deux dimensions de l'espace latent sont ici illustrées : l'une influe sur l'orientation de la tête, l'autre sur les émotions du visage.



Image de [Kingma et al.] Auto-encoding variational Bayes

## Entraînement des auto-encodeurs



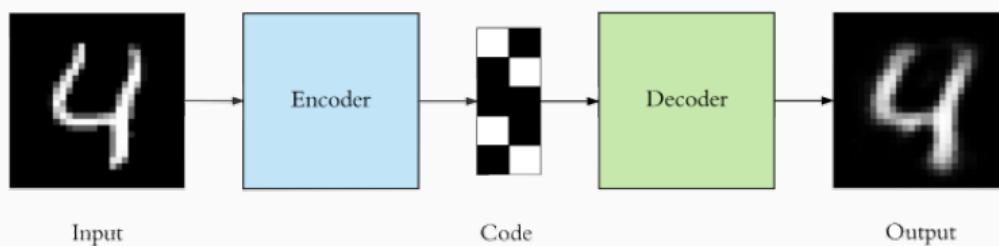
- Fonction d'**activation linéaire** sur la couche de sortie,
- Fonction de **coût quadratique** pour des images,  
entropie-croisée pour des données catégorielles,
- Fonction objectif à minimiser :

$$J := \sum_{i=1}^n \|x^{(i)} - \hat{x}^{(i)}\|^2$$

# Auto-encodeurs et Régularisation

***En pratique :*** Les auto-encodeurs sont *difficiles à entraîner/calibrer...*

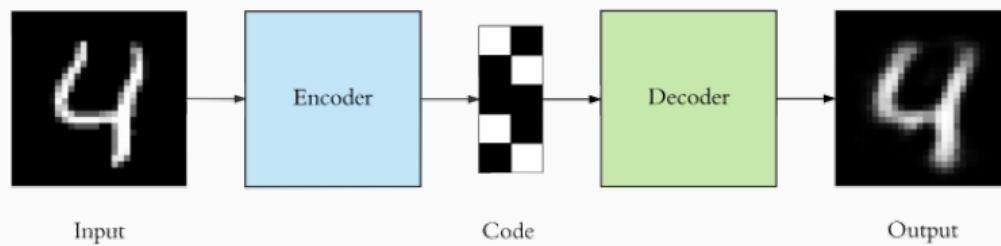
- **L'espace latent** doit avoir une dimension semblable à la variété sur laquelle vivent les données ;  
Il est malheureusement impossible de connaître cette dimension à l'avance.
- **L'encodeur** (et le **décodeur**) doit avoir une capacité suffisante pour apprendre la fonction qui va de l'espace des données vers l'espace latent (et son inverse).



# Auto-encodeurs et Régularisation

**En pratique :** Les auto-encodeurs sont *difficiles à entraîner/calibrer...*

- **L'espace latent** doit avoir une dimension semblable à la variété sur laquelle vivent les données ;  
Il est malheureusement impossible de connaître cette dimension à l'avance.
- **L'encodeur** (et le **décodeur**) doit avoir une capacité suffisante pour apprendre la fonction qui va de l'espace des données vers l'espace latent (et son inverse).



~~> Comme pour les réseaux de neurones classiques, il peut être intéressant de procéder à une **régularisation** des auto-encodeurs pour en améliorer l'entraînement.

## Auto-encodeurs épars ou Sparse autoencoder

On appelle auto-encodeur épars (**sparse auto-encoder**) un auto-encodeur que l'on entraîne en optimisant la fonction objectif suivante (*Pénalisation LASSO*) :

$$J = \sum_{i=1}^n \|x^{(i)} - \hat{x}^{(i)}\|^2 + \lambda|z|$$

Cette régularisation constraint les variables de l'espace latent : *Seules un petit nombre d'entre elles peuvent être actives à la fois.*

---

## Auto-encodeurs épars ou Sparse autoencoder

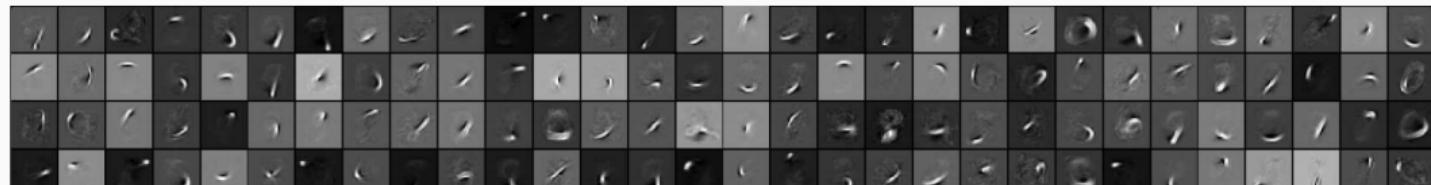
On appelle auto-encodeur épars (**sparse auto-encoder**) un auto-encodeur que l'on entraîne en optimisant la fonction objectif suivante (*Pénalisation LASSO*) :

$$J = \sum_{i=1}^n \|x^{(i)} - \hat{x}^{(i)}\|^2 + \lambda|z|$$

Cette régularisation constraint les variables de l'espace latent : *Seules un petit nombre d'entre elles peuvent être actives à la fois.*

---

Les auto-encodeurs épars se sont avérés utile pour l'**apprentissage de caractéristiques** qui peuvent être ré-utilisées pour la classification.



## Auto-encodeur épars et Recherche d'information

On appelle auto-encodeur épars (**sparse auto-encoder**) un auto-encodeur que l'on entraîne en optimisant la fonction objectif suivante (*Pénalisation LASSO*) :

$$J = \sum_{i=1}^n \|x^{(i)} - \hat{x}^{(i)}\|^2 + \lambda|z|$$

Cette régularisation constraint les variables de l'espace latent : *Seules un petit nombre d'entre elles peuvent être actives à la fois.*

---

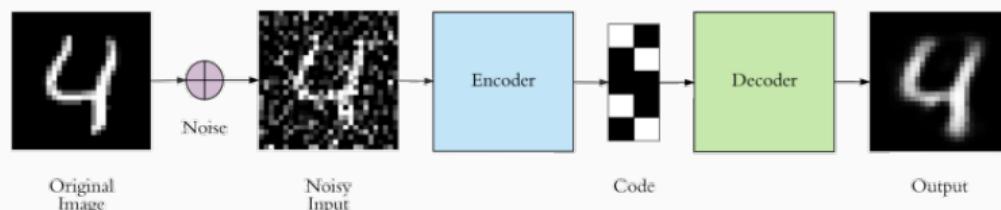
En particulier, on peut forcer l'espace latent à adopter une **représentation épars binaire**, en affectant une fonction d'**activation sigmoïde** à la couche latente de l'auto-encodeur.

~~> Utile pour la **recherche d'information** : Permet de calculer la distance entre la représentation de deux données avec un produit scalaire de deux vecteurs creux, ce qui est très rapide.

## Autre technique de régularisation : Ajout de bruit aléatoire

**Idée** : Forcer l'auto-encodeur à apprendre une **représentation intelligente**, i.e. des caractéristiques utiles par ajout de **bruit aléatoire** à ses entrées.

~~> L'auto-encodeur ne peut plus “simplement” copier son entrée sur sa sortie, il lui faut soustraire le bruit et retrouvé les **données sous-jacentes significatives**.



~~> **Auto-codeur débruiteur** ou de débruitage

# Plan du cours

Introduction et Motivation

Auto-encodeurs : Premières définitions

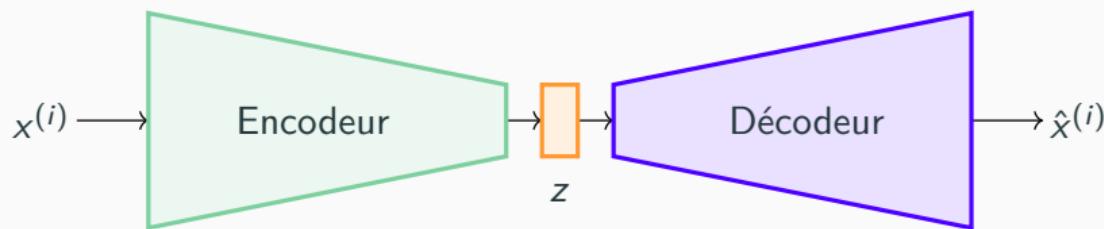
Auto-encodeurs : Entrainement et Calibration

Auto-encodeurs : Exemples d'application

Apprentissage auto-supervisé

## Auto-encodeur et Transfert d'apprentissage ↽ Classification

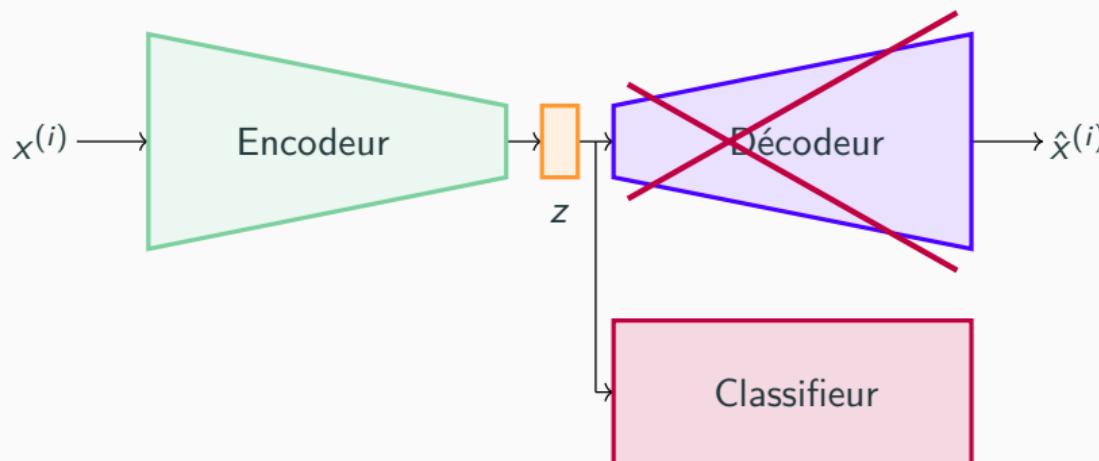
**Question :** Comment ré-utiliser l'espace latent en classification ?



# Auto-encodeur et Transfert d'apprentissage $\rightsquigarrow$ Classification

**Question :** Comment ré-utiliser l'espace latent en classification ?

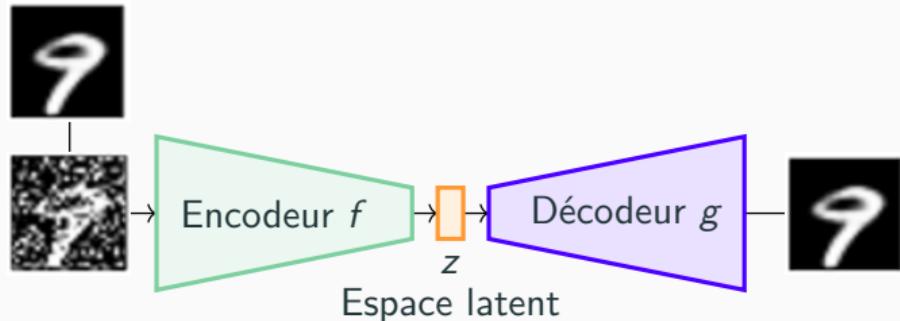
Grâce au *transfert learning* !



1. Entraînement de l'auto-encodeur sur une large base de données *non annotée*.
2. **Fine-tuning** du classifieur sur une petite base de données *annotée*.

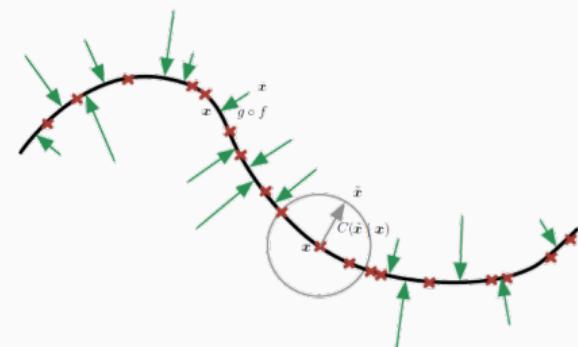
$\rightsquigarrow$  Une forme d'**apprentissage semi-supervisé**.

## Auto-encodeur débruiteur

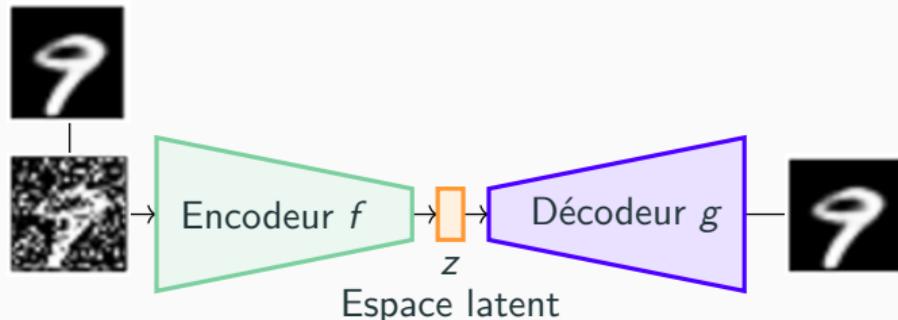


Un auto-encodeur débruiteur tend à ramener une donnée bruitée vers la variété qui décrit les données.

L'auto-encodeur décrit ainsi un champ de vecteurs pointant sur la variété.



## Auto-encodeur débruiteur



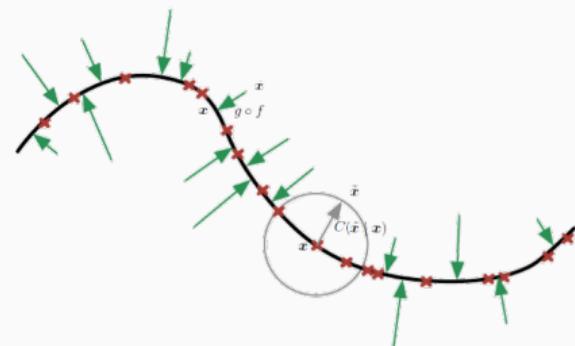
Un auto-encodeur débruiteur tend à ramener une donnée bruitée vers la variété qui décrit les données.

L'auto-encodeur décrit ainsi un champ de vecteurs pointant sur la variété.

Si on nomme  $f$  (resp.  $g$ ) la fonction décrite par l'encodeur (resp. le décodeur), un auto-encodeur débruiteur cherche à optimiser la fonction objectif suivante :

$$J := \sum_{i=1}^n \|x^{(i)} - g \circ f(\tilde{x}^{(i)})\|^2$$

où  $\tilde{x}^{(i)}$  correspond à une version bruitée de  $x^{(i)}$ .



# Auto-encodeur débruiteur – Exemples d'applications

## Débruitage de signal sonore :

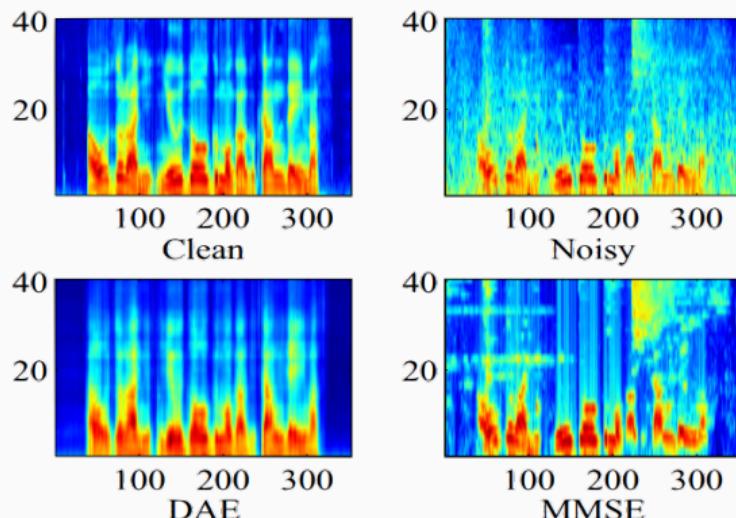


Image de [Lu et al.]

Speech Enhancement Based on Deep Denoising Autoencoder

## Inpainting :

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A learner

In machine learning, unsupervised learning refers to the problem of

Since the end of the age of sail a ship has been any large buoyant

Since the end of the age of sail a ship has been

Since the end of the age of sail a ship has been

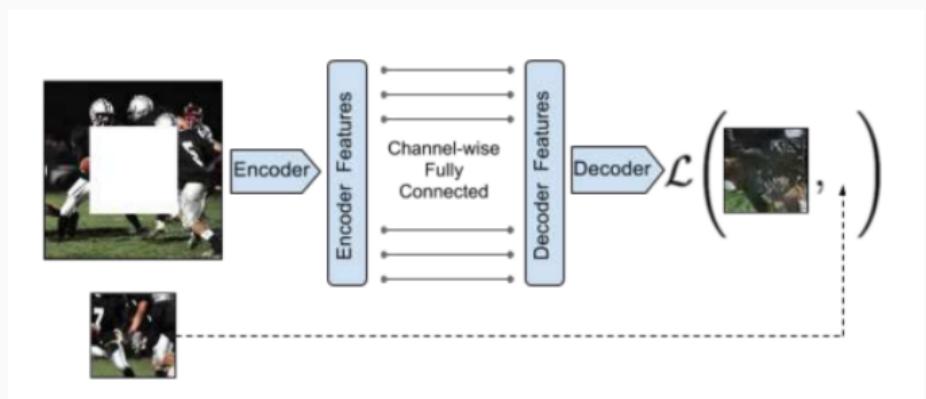
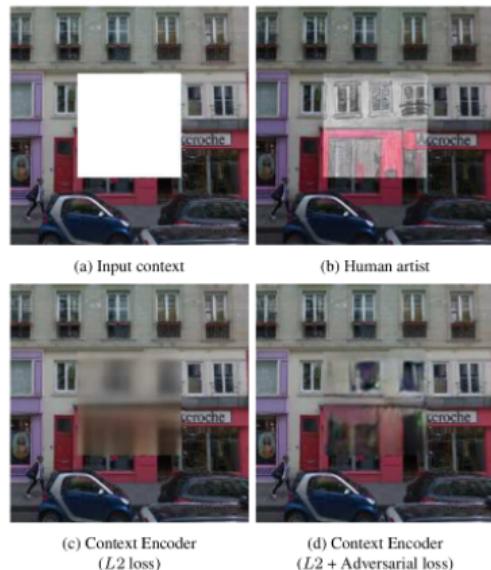
Since the end of the age of sail a ship has been any NOISY



Image de [Xie et al.]

Image Denoising and Inpainting with Deep Neural Networks

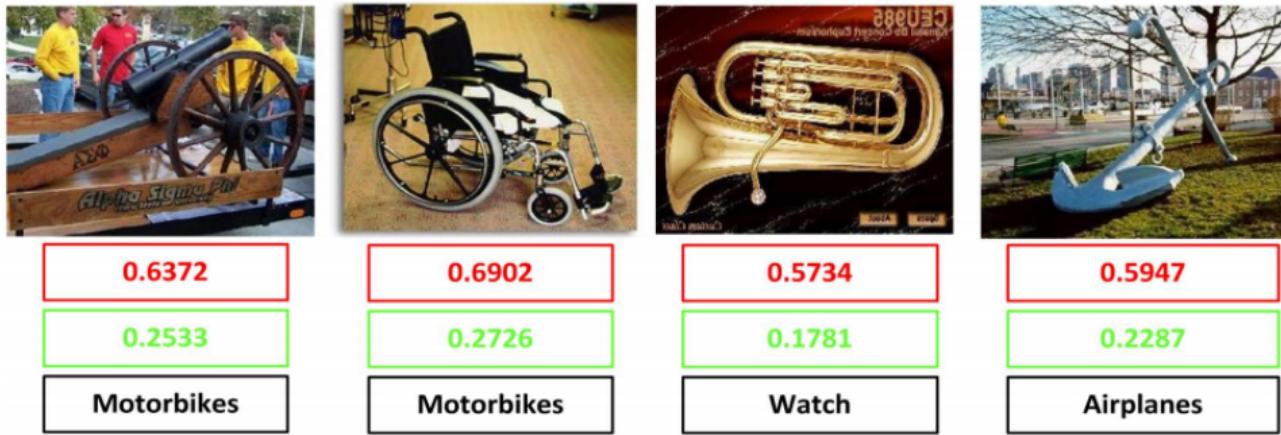
## Une variante : Auto-codeurs contextuels ou *Context autoencoders*



On parle pour cette méthode d'**apprentissage auto-supervisé** (*self-supervised learning*),  
c'est-à-dire d'un apprentissage où la supervision est assurée par la donnée elle-même.

# Auto-encodeur et Détection d'anomalies

**Exemple :** Détection d'anomalies dans la base de données Caltech-101 :



La détection d'anomalies a de nombreuses applications en sécurité, réseau, vidéo-surveillance, maintenance, etc.

## Bilan sur les auto-encodeurs

- Les auto-encodeurs forment une classe de réseaux de neurones permettant de faire de l'**apprentissage non supervisé**, ou parfois semi-supervisé.
- Ils sont aussi très utiles pour faire de l'**analyse de données**, et de la fouille de données, par leur capacité à révéler des variables structurant les données.
- Ils sont utilisés dans de nombreuses applications, et sont d'excellents exemples d'utilisation pratique de l'apprentissage non-supervisé.

# Plan du cours

Introduction et Motivation

Auto-encodeurs : Premières définitions

Auto-encodeurs : Entrainement et Calibration

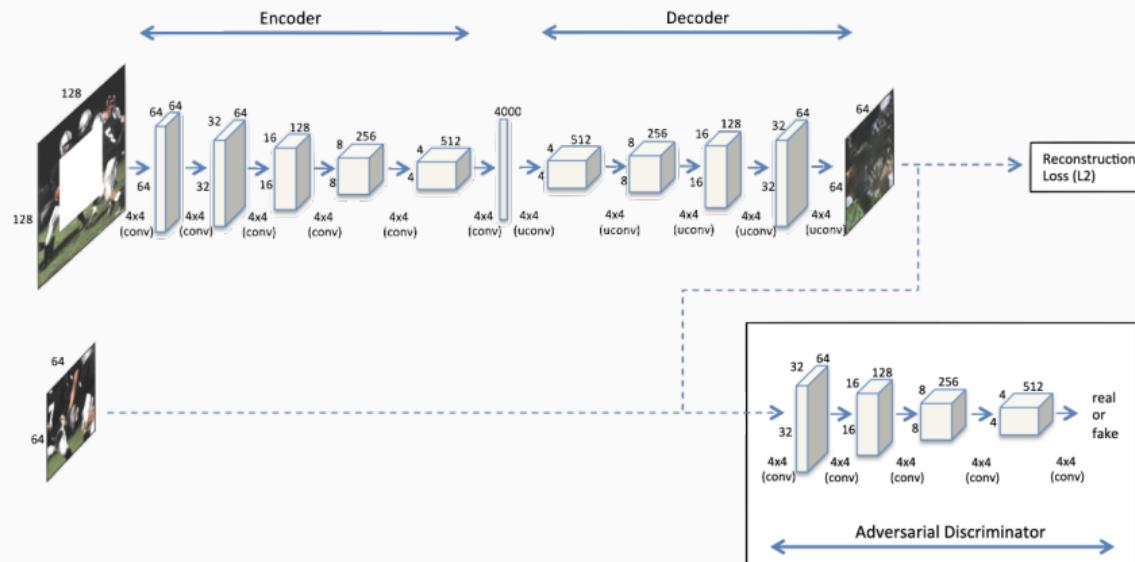
Auto-encodeurs : Exemples d'application

Apprentissage auto-supervisé

# Apprentissage auto-supervisé

On parle d'**apprentissage auto-supervisé** (*self-supervised learning*) lorsque la supervision est assurée par la donnée elle-même.

**Exemple** : Auto-codeurs contextuels, Auto-encodeurs débruiteurs.



## Apprentissage auto-supervisé

On peut regrouper les algorithmes d'apprentissage auto-supervisé en *trois familles* de méthodes :

1. Les méthodes qui cherchent à **prédir ou restaurer** une partie de la donnée à partir d'une autre partie,

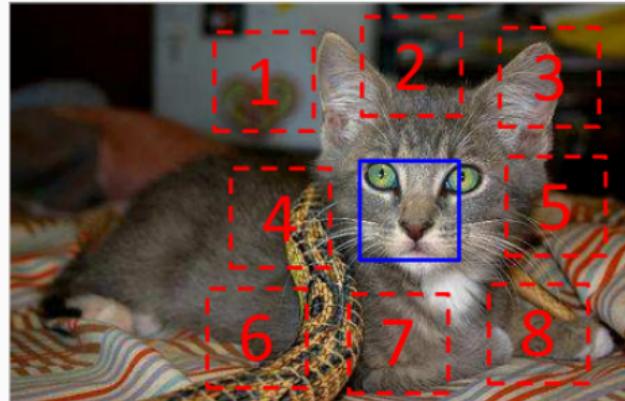
Exemples : Auto-codeurs contextuels, Auto-encodeurs débruiteurs, etc.

2. Les méthodes qui cherchent à **déceler des caractéristiques élémentaires** des données,

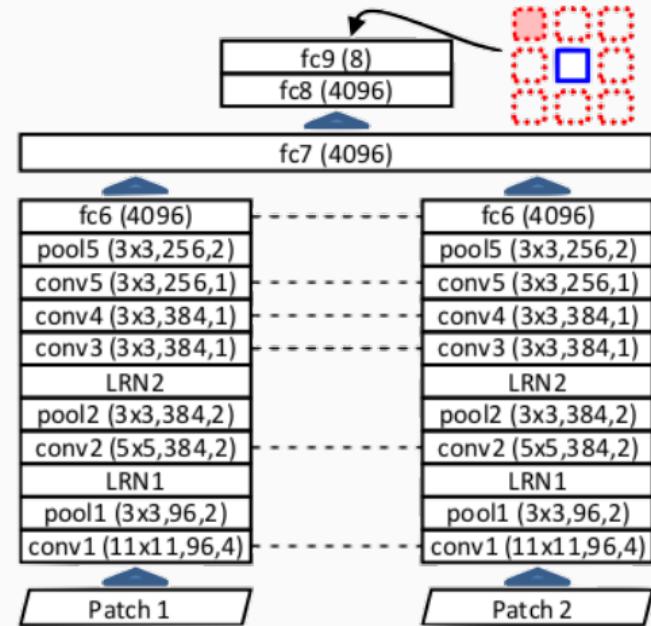
Exemple : Apprentissage non supervisé de représentations visuelles par prédiction du contexte, ou de l'orientation.

3. Les méthodes **contrastives**.

# Apprentissage par prédiction du contexte

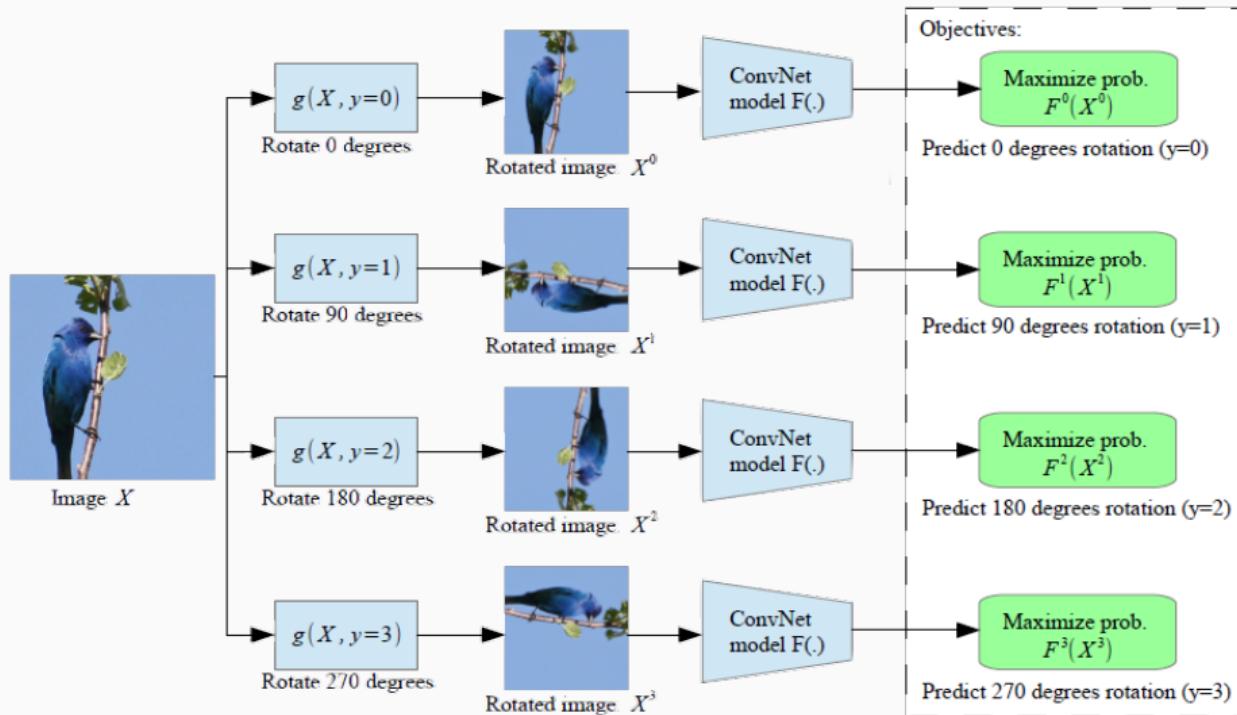


$$X = (\text{Patch 1}, \text{Patch 2}); Y = 3$$



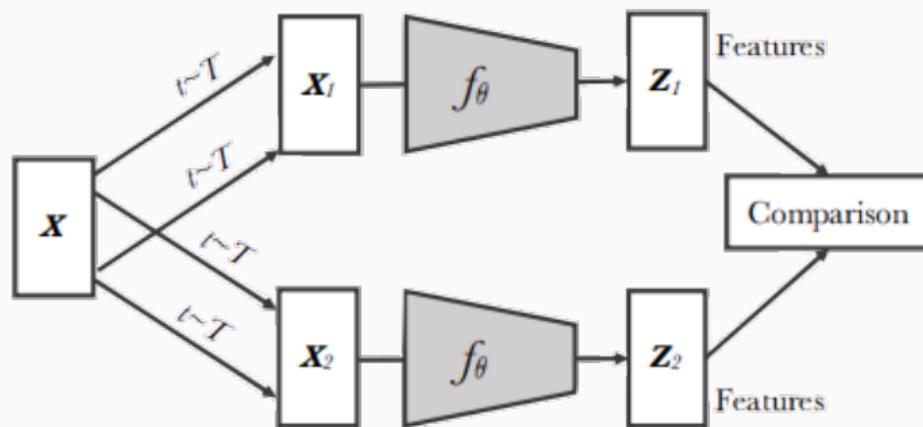
**Objectif** : Prédiction de la **position relative** de deux patches extraits de l'image.

# Apprentissage par prédition de l'orientation



## Apprentissage contrastif standard (SimCLR)

Utilisation de **réseaux siamois** pour apprendre des descripteurs  $z_i$  proches pour des images similaires et éloignés pour des images différentes.



## Apprentissage contrastif par cluster (SwAV)

Avec une idée similaire, cette méthode passe par un **clustering intermédiaire** dans l'espace de descripteurs et cherche à assigner le même cluster à des vues (augmentations) différentes de la même image.

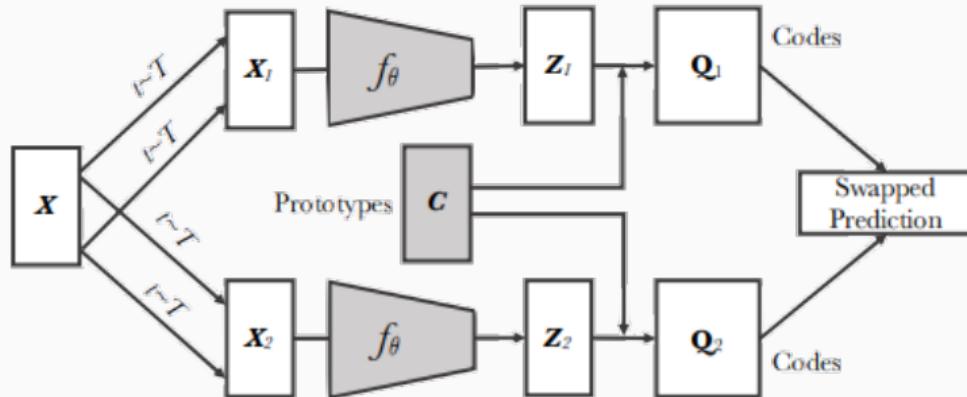


Image de [Caron et al.] Unsupervised Learning of Visual Features by Contrasting Cluster Assignments.

# Classification auto-supervisée sur ImageNet

Rank	Model	Top 1 Accuracy	Number of Params	Extra Training Data	Paper	Code	Result	Year	Tags	if					
1	iBOT (ViT-L/16)	82.3%	307M	✓	iBOT: Image BERT Pre-Training with Online Tokenizer	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer colab					<small>Contrastive IO-IB RegNet</small>	
2	EsViT (Swin-B)	81.3%	87M	✗	Efficient Self-supervised Vision Transformers for Representation Learning	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer					<small>Transformer</small>	
3	iBOT (ViT-L/16)	81.3%	307M	✗	iBOT: Image BERT Pre-Training with Online Tokenizer	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer						
4	MoCo v3 (ViT-BN-L/7)	81.0%	304M	✗	An Empirical Study of Training Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer Contrastive						
5	DINO (XCiT-M24/8/384)	80.9%	84M	✗	XCiT: Cross-Covariance Image Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer						
6	EsViT (Swin-5)	80.8%	49M	✗	Efficient Self-supervised Vision Transformers for Representation Learning	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer						
7	RELICv2 (ResNet-200 x2)	80.6%		✗	Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet?		<a href="#">🔗</a>	2022							
8	DINO (XCiT-M24/8)	80.3%	84M	✗	XCiT: Cross-Covariance Image Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer						
9	DINO (ViT-B/8)	80.1%	85M	✗	Emerging Properties in Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	Transformer						
10	SimCLRv2 (ResNet-152 x3, 5K)	79.8%	795M	✗	Big Self-Supervised Models are Strong Semi-Supervised Learners	<a href="#">🔗</a>	<a href="#">🔗</a>	2020	Contrastive ResNet						
11	SEER (RegNet10B)							79.8%	10000M	✓	Vision Models Are More Robust And Fair When Pretrained On Uncurated Images Without Supervision	<a href="#">🔗</a>	<a href="#">🔗</a>	2022	<small>Contrastive IO-IB RegNet</small>
12	DINO (ViT-S/8)							79.7%	21M	✗	Emerging Properties in Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	<small>Transformer</small>
13	BYOL (ResNet-200 x2)							79.6%	250M	✗	Bootstrap your own latent: A new approach to self-supervised Learning	<a href="#">🔗</a>	<a href="#">🔗</a>	2020	<small>Contrastive ResNet</small>
14	iBOT (ViT-B/16)							79.5%		✗	iBOT: Image BERT Pre-Training with Online Tokenizer	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	
15	DINO (XCiT-512/8)							79.2%	26M	✗	XCiT: Cross-Covariance Image Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	<small>Transformer</small>
16	MoCo v3 (ViT-BN-H)							79.1%		✗	An Empirical Study of Training Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	<small>Transformer Contrastive</small>
17	C-BYOL (ResNet-50 2x, 1000 epochs)							78.8%		✗	Compressive Visual Representations	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	
18	DINO (XCiT-M24/16)							78.8%	84M	✗	XCiT: Cross-Covariance Image Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	
19	BYOL (ResNet-50 x4)							78.6%	375M	✗	Bootstrap your own latent: A new approach to self-supervised Learning	<a href="#">🔗</a>	<a href="#">🔗</a>	2020	<small>Contrastive ResNet</small>
20	SwAV (ResNet-50 x4)							78.5%	586M	✗	Unsupervised Learning of Visual Features by Contrasting Cluster Assignments	<a href="#">🔗</a>	<a href="#">🔗</a>	2020	<small>Contrastive ResNet</small>
21	DINO (ViT-B/16)							78.2%	85M	✗	Emerging Properties in Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	<small>Transformer</small>
22	MoCo v3 (ViT-H)							78.1%	632M	✗	An Empirical Study of Training Self-Supervised Vision Transformers	<a href="#">🔗</a>	<a href="#">🔗</a>	2021	<small>Transformer Contrastive</small>

Le réseau RegNetY (**1.3 milliards de paramètres**), pré-entraîné (sur 512 GPU) de manière auto-supervisée par l'algorithme SwAV sur un milliard d'images extraites aléatoirement d'Instagram, et spécifié (*fine-tuned*) sur ImageNet obtient 84.2 % de précision top-1 sur cette base, soit la meilleure performance à ce jour.

[Goyal et al.] Self-supervised Pretraining of Visual Features in the Wild.

