Auto-encodeurs et modèles génératifs

IA et Multimédia

A. Carlier

2020

Cours précédents

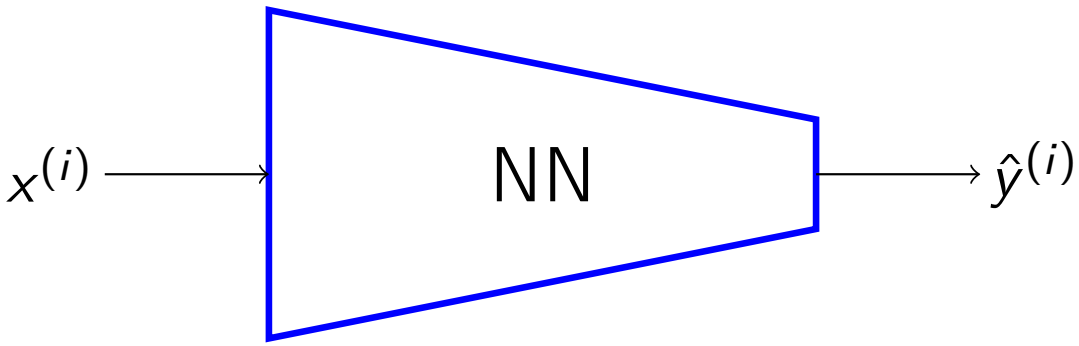
- Réseaux de neurones
- Réseaux de neurones convolutifs (applications à l'image)
- Réseaux de neurones récurrents (applications au texte et à l'audio)

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 le note :

$$\mathcal{D} = \{(\mathrm{x}^{(1)}, y^{(1)}), .., (\mathrm{x}^{(m)}, y^{(m)})\}.$$

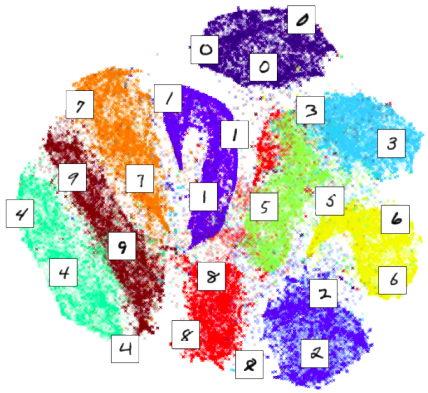
Les labels permettent d'enseigner à l'algorithme à établir des correspondances entre les observations et les labels.



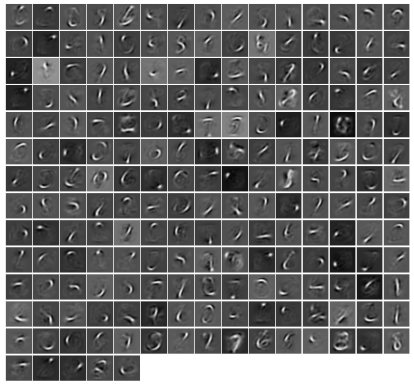
Apprentissage non-supervisé

Dans le cadre de l'**apprentissage non supervisé**, on dispose uniquement d'observations

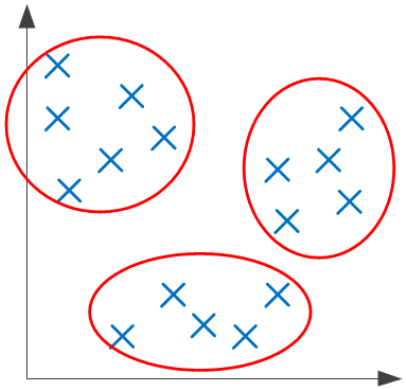
$$\mathcal{D} = \{\mathrm{x}^{(1)}, .., \mathrm{x}^{(m)}\}.$$



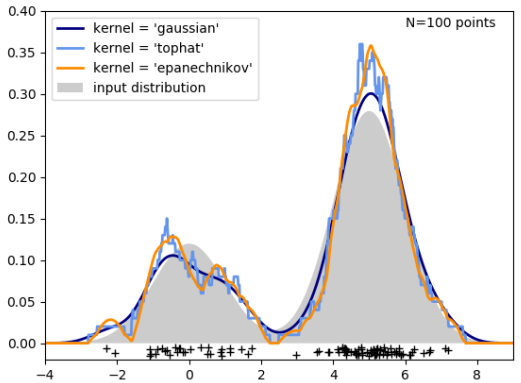
Réduction de dimension



Extraction de caractéristiques



Clustering

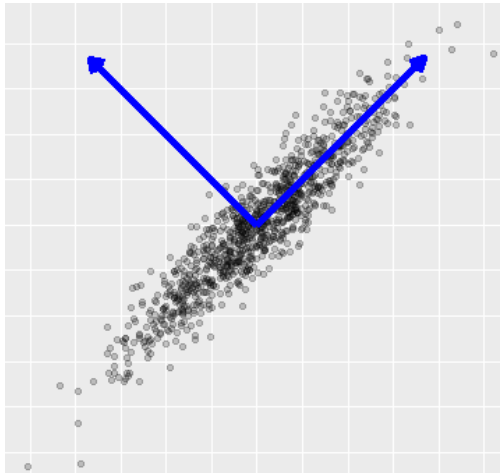


Estimation de densité

Réduction de dimension : ACP

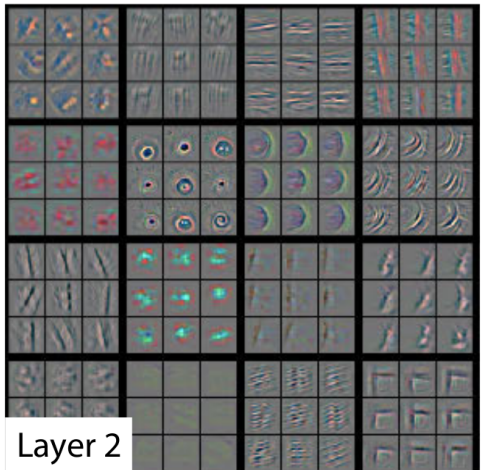
L'Analyse en Composantes Principales est une technique de **réduction de dimension** dans laquelle on cherche à projeter des données dans un sous-espace de plus faible dimension en maximisant un critère "d'étalement" des données.

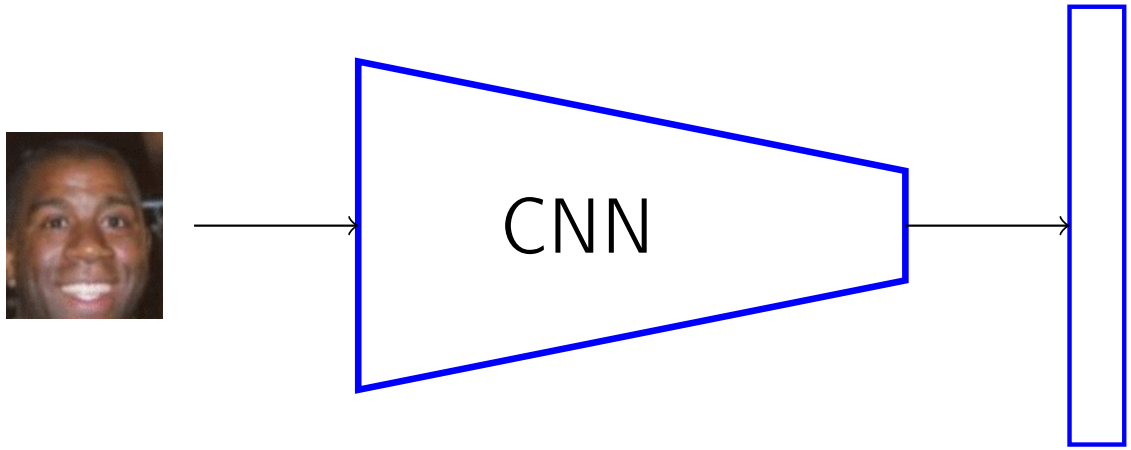
La méthode s'appuie sur une diagonalisation de la matrice de variance-covariance. La base de l'espace de projection est obtenue à partir des vecteurs propres associés aux plus grandes valeurs propres.



Extraction de caractéristiques

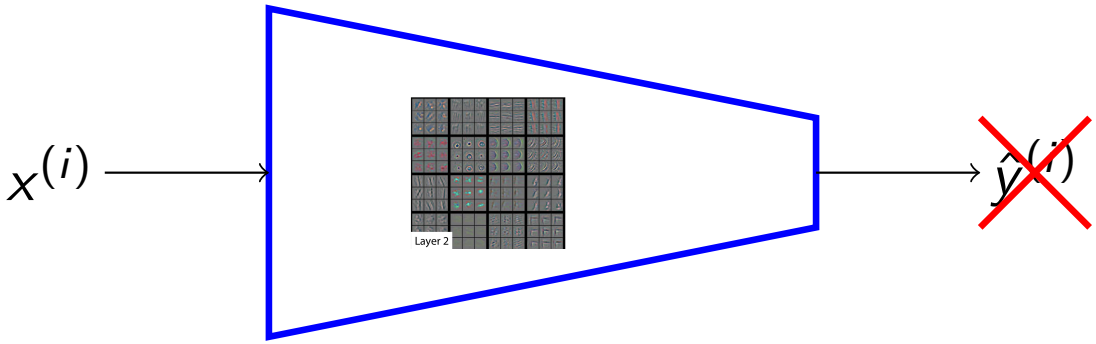
Qu'entend-on par **extraction de caractéristiques** ? Nous en avons déjà vu des exemples avec l'interprétation des filtres appris par le réseau AlexNet, ou encore par le réseau DeepFace pour la reconnaissance de visage.





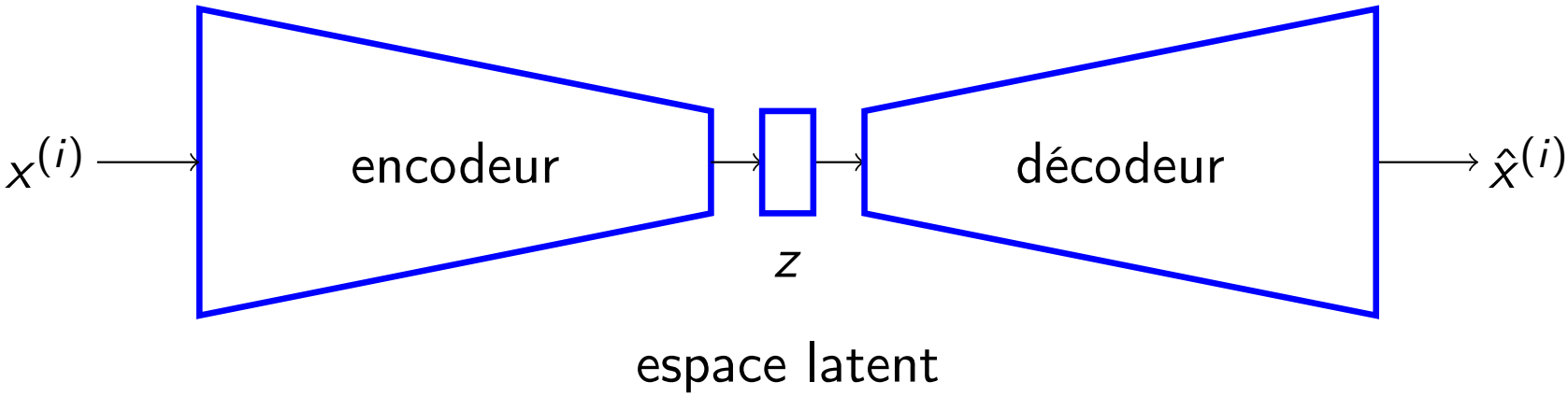
Apprentissage non-supervisé

Comment extraire des caractéristiques, semblables à celles apprises par AlexNet, mais sans annotations ?

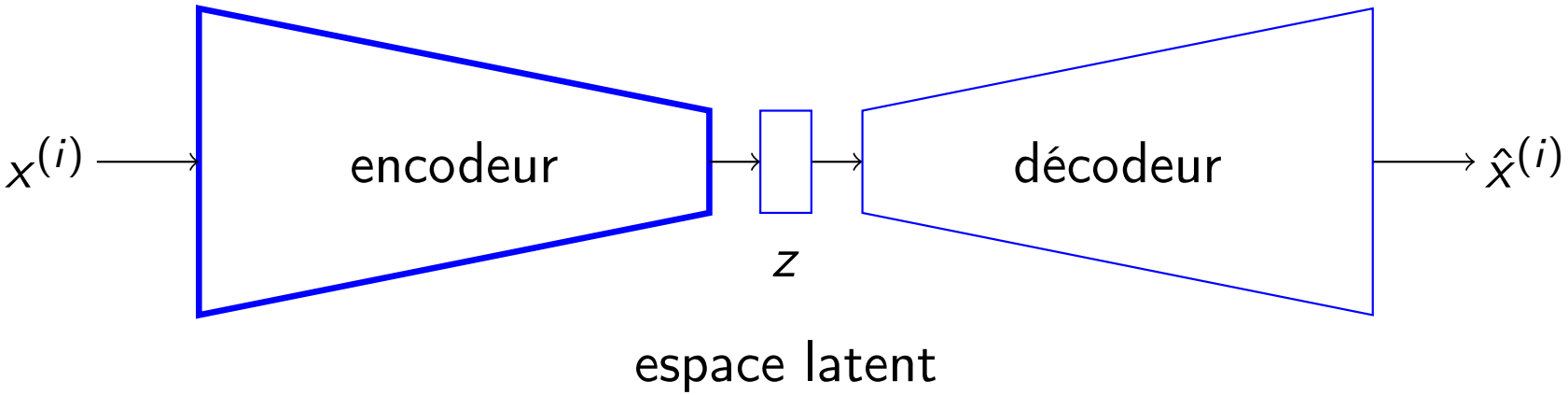


Auto-encodeurs

Solution : chercher à prédire les données d'entrée !



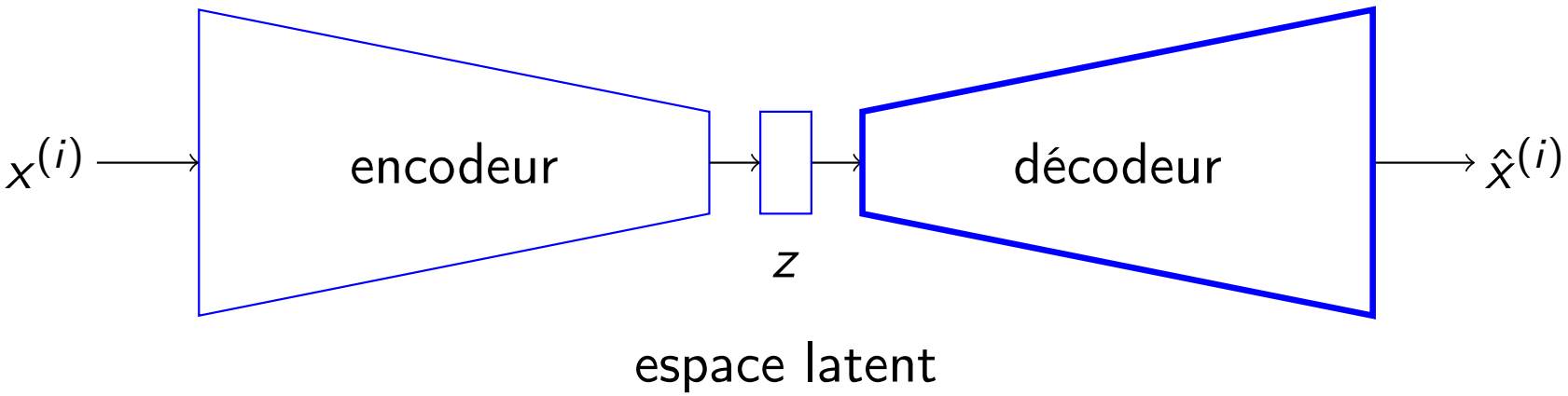
Auto-encodeurs



L'**encodeur** :

- Extrait les caractéristiques principales (structure) de l'entrée.
- Compresse 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

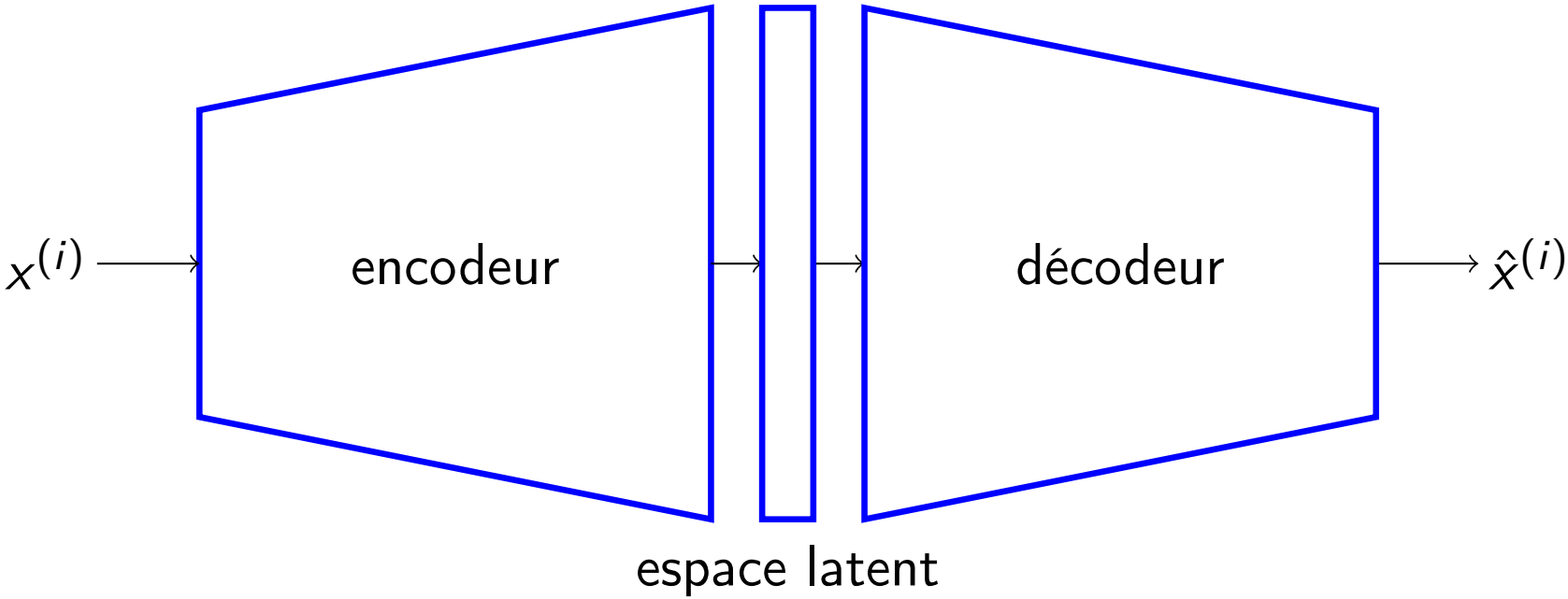


Le **décodeur** :

- Reconstruit une approximation de l'entrée initiale à partir de sa représentation latente.
- Est une forme de modèle génératif (cf. seconde partie de ce cours).

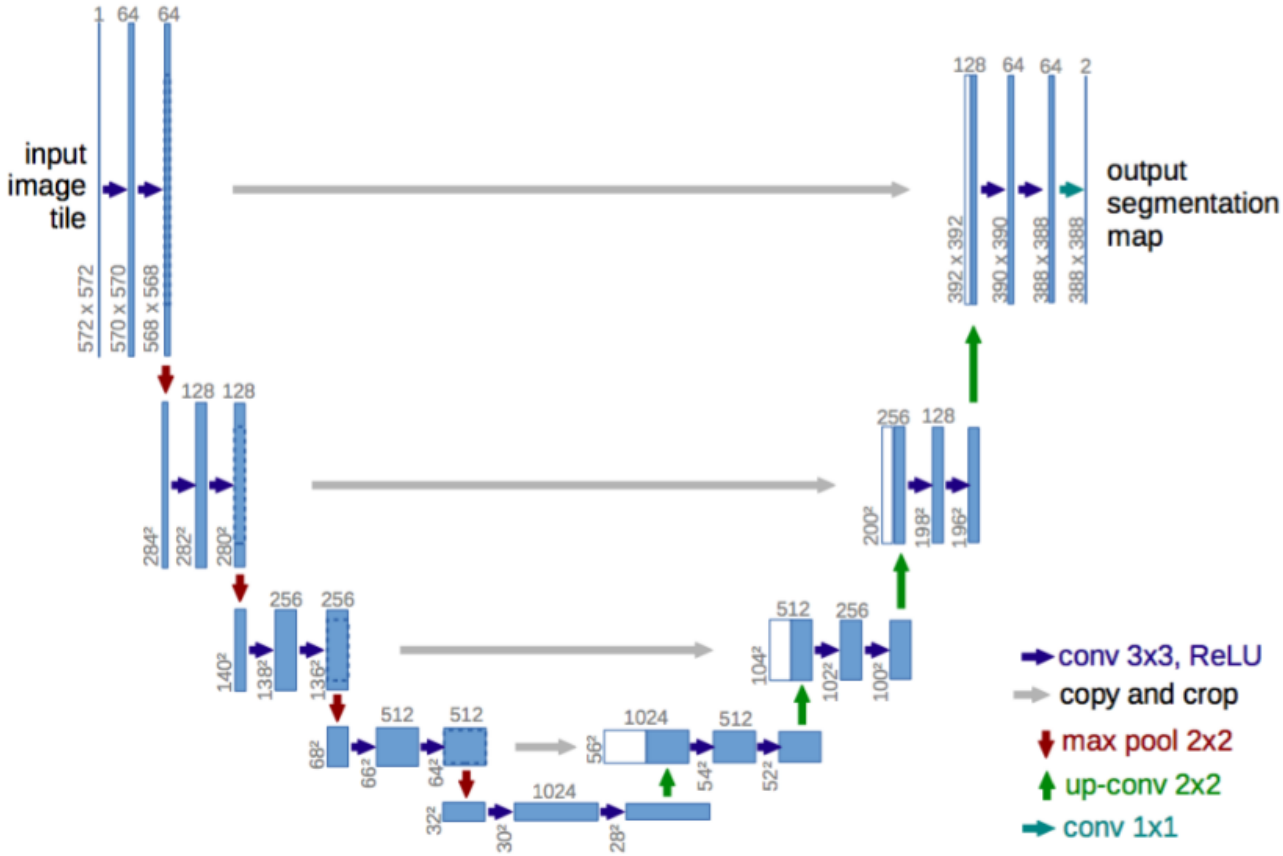
Auto-encodeurs

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.

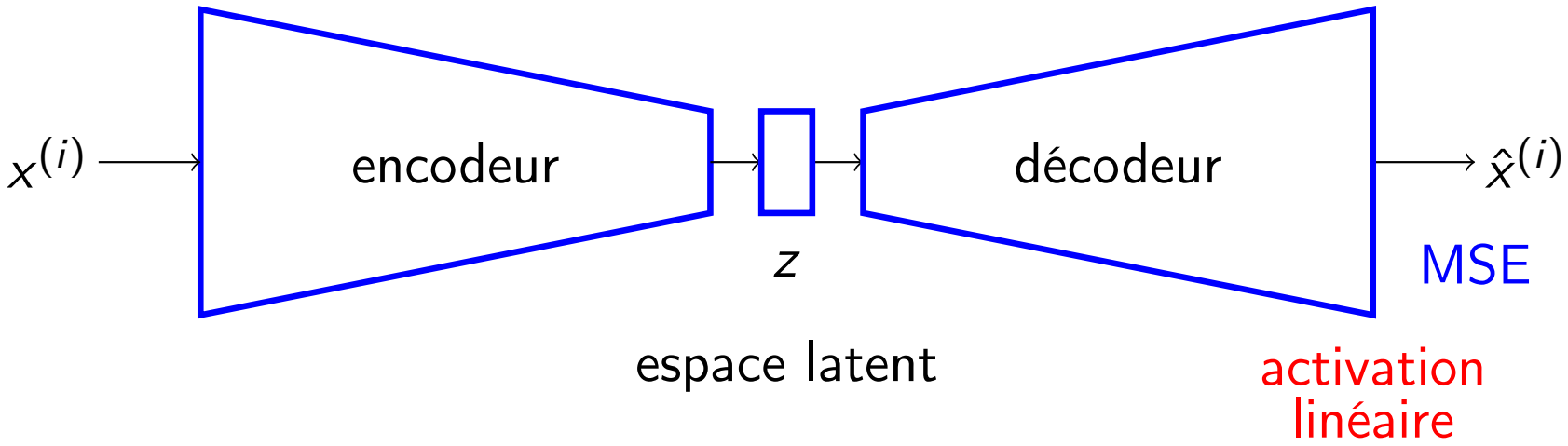
Remarque : UNet



Le réseau UNet peut, en altérant légèrement sa forme décrite ci-dessus, être vu comme un auto-encodeur convolutionnel.

[Ronneberger et al.] U-Net : Convolutional Networks for Biomedical Image Segmentation.

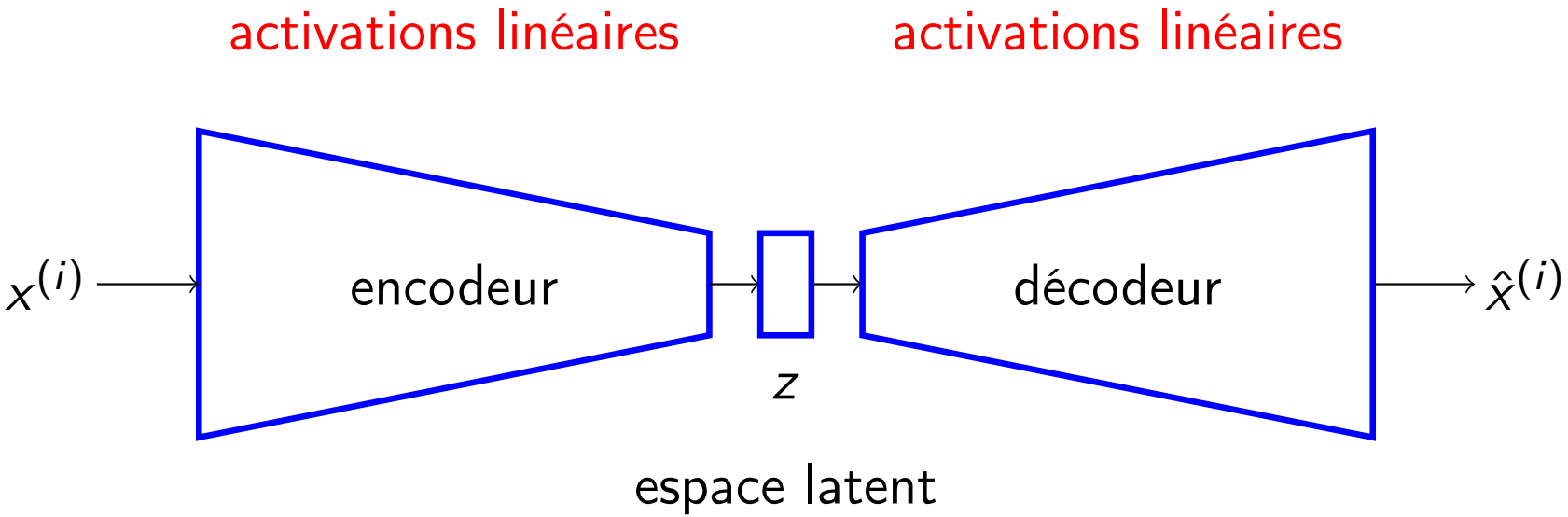
Entraînement d'un auto-encodeur



- Fonction d'activation linéaire sur la couche de sortie.
- Fonction de coût quadratique.
- Fonction objectif à minimiser :

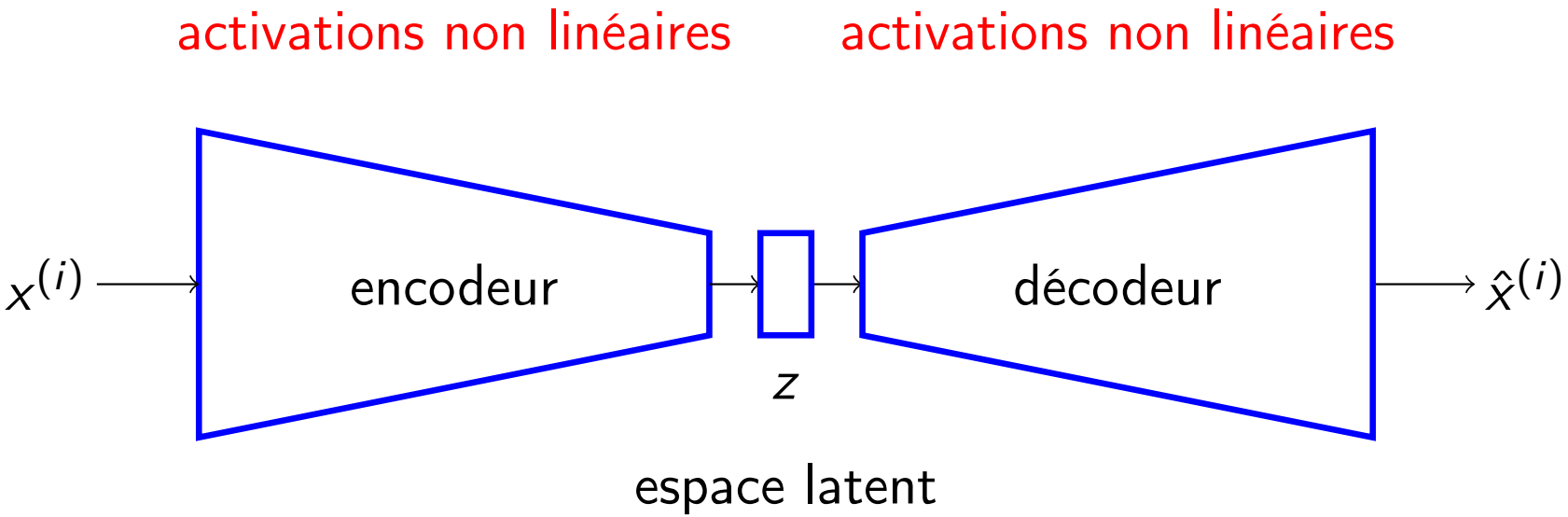
$$J = \sum_{i=1}^{n} ||x^{(i)} - \hat{x}^{(i)}||^2$$

Auto-encodeur pour la réduction de dimension



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

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.

Variété

Les variétés sont des espaces généralisant les courbes (dimension 1) et surfaces (dimension 2) à des dimensions supérieures.

Les données que nous étudions en Multimédia sont en général, dans des espaces de très haute dimension, concentrées autour de variétés (exemple ci-dessous).

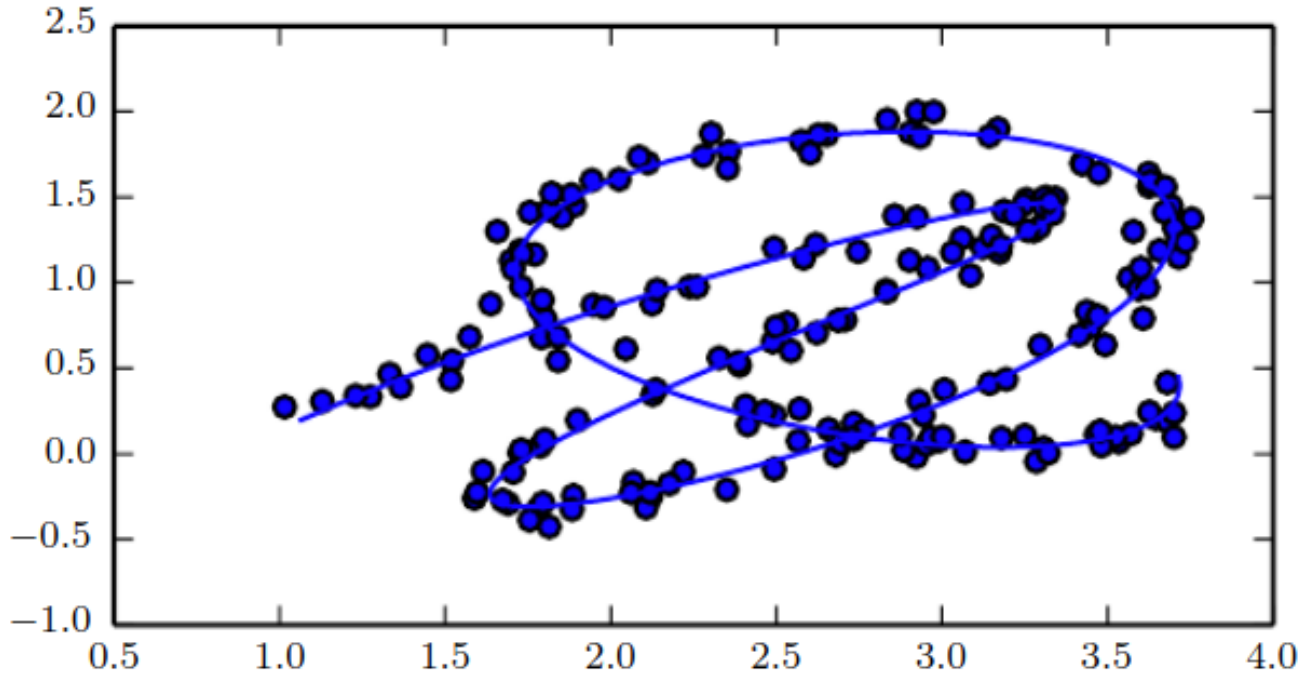


Image de [Goodfellow et al.] Deep Learning

Variétés et auto-encodeurs

On cherche à décrire, dans l'espace latent, la variété sur laquelle vivent nos données, i.e. un sous-espace sur lequel on a une haute probabilité de rencontrer des données d'apprentissage.

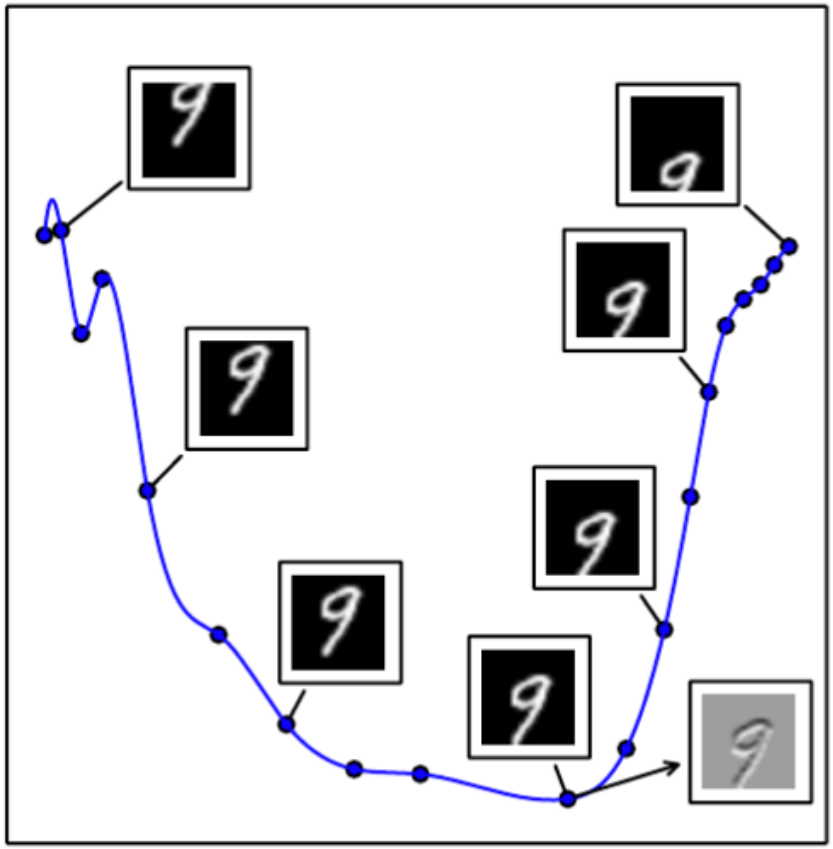


Image de [Goodfellow et al.] Deep Learning

Variétés et auto-encodeurs

Dans l'exemple ci-dessous, deux dimensions de l'espace latent sont 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

Auto-encodeurs et régularisation

Les auto-encodeurs sont difficiles à entraîner !

- 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).

$\rightarrow$ Comme pour les réseaux de neurones classiques, il est intéressant de procéder à une régularisation des auto-encodeurs pour en améliorer l'entraînement.

Auto-encodeur épars

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

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

Cette régularisation contraint 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.

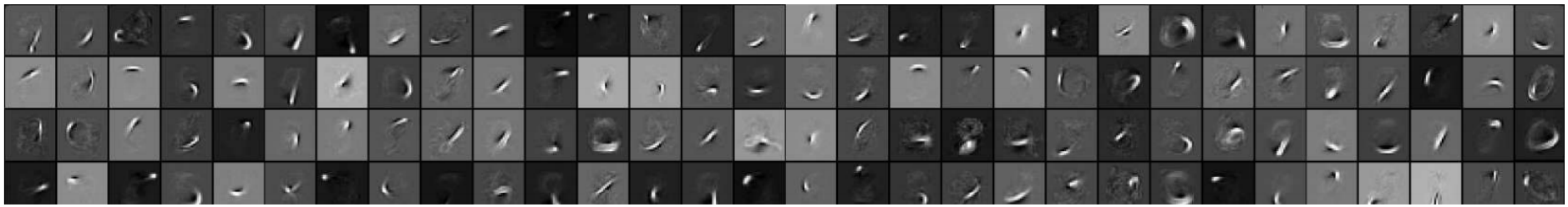
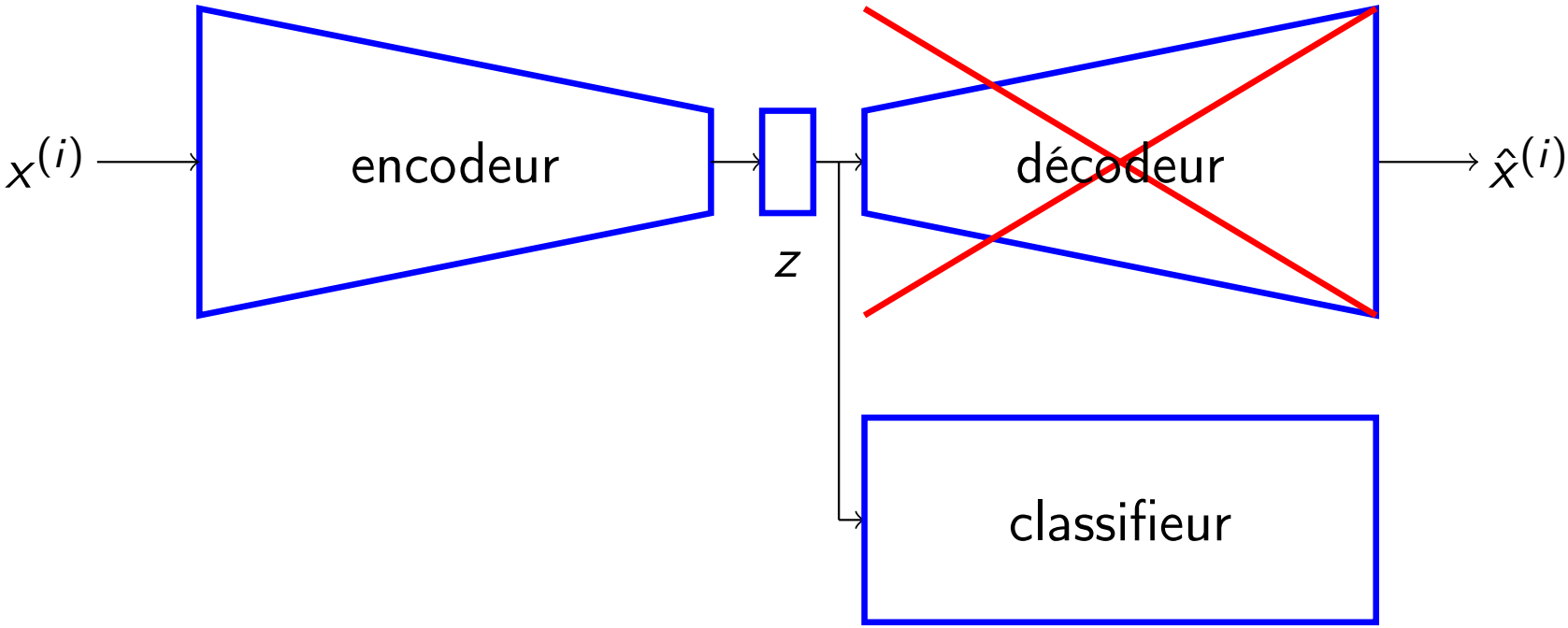


Image de [Makhzani et al.] k-sparse autoencoders

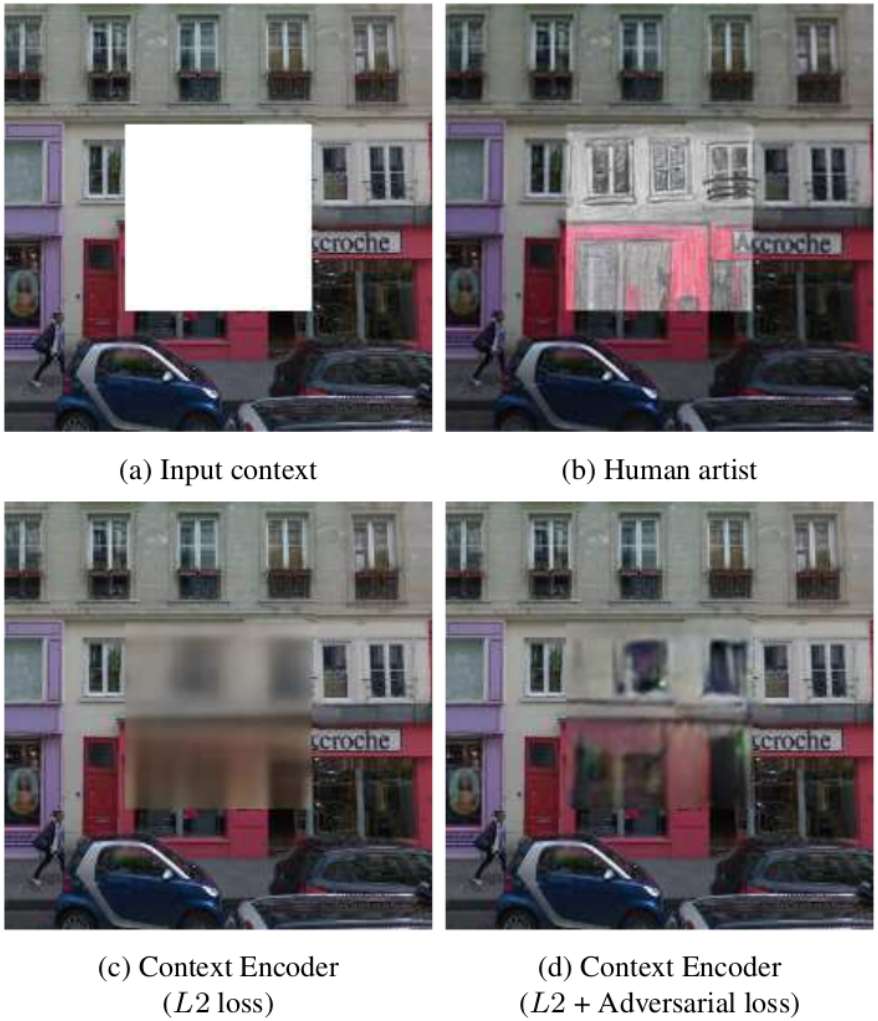
Auto-encodeur et transfer learning



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.

$\rightarrow$ une forme d'**apprentissage semi-supervisé** !

Une variante : *Context autoencoders*



(a) Input context (b) Human artist

(c) Context Encoder ($L2$ loss) (d) Context Encoder ($L2$ + Adversarial loss)

Image de [Pathak et al.] Context Encoders : Feature Learning by Inpainting.

Une variante : *Context autoencoders*

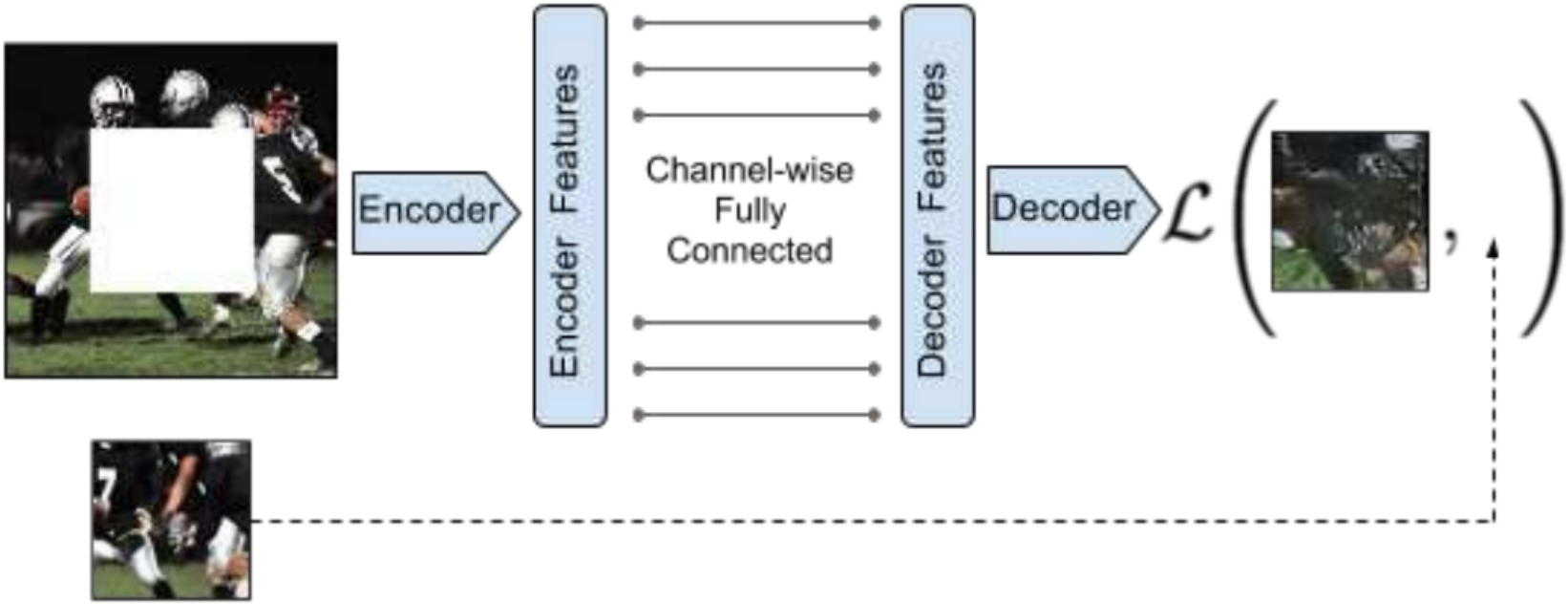


Image de [Pathak et al.] Context Encoders : Feature Learning by Inpainting.

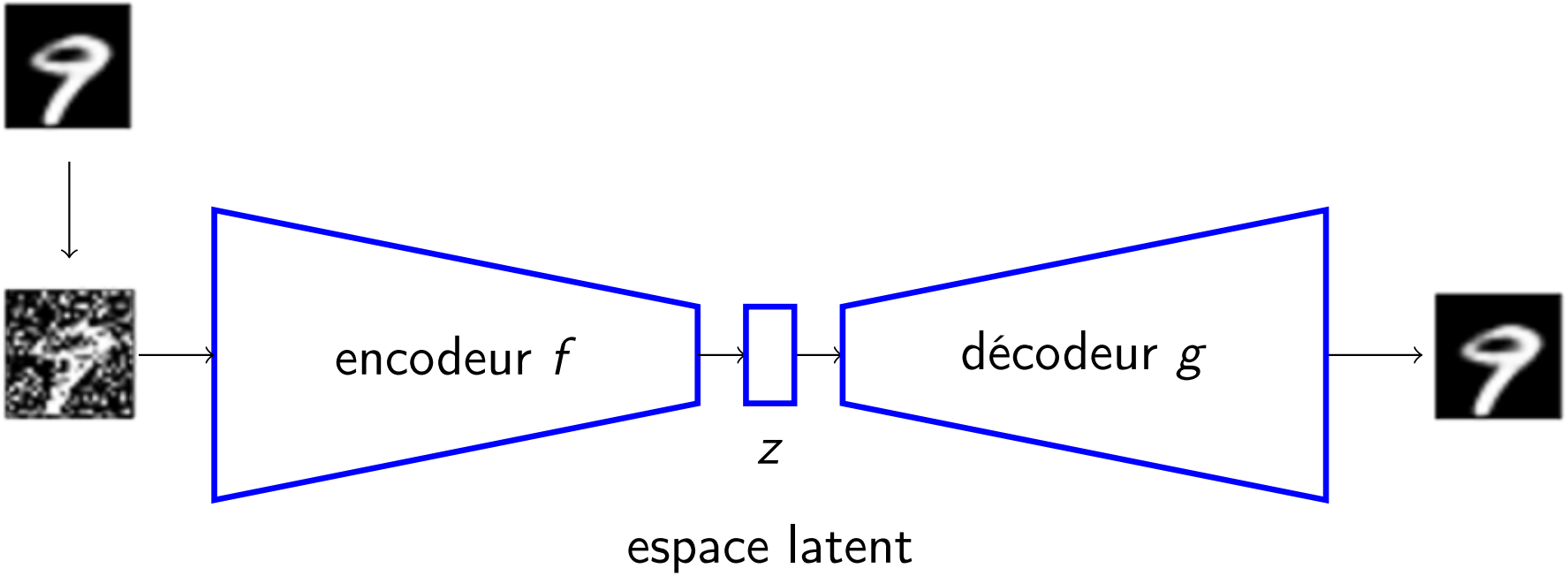
Auto-encodeur épars et recherche d'information

On peut forcer l'espace latent à adopter une représentation éparse **binaire**, en affectant une fonction d'activation sigmoide à la couche latente de l'auto-encodeur.

Cette propriété est désirable en **recherche d'information**, car elle permet de calculer la distance entre la représentation de deux données avec un produit scalaire de deux vecteurs creux, ce qui est très rapide.

[Salakhutdinov et al.] Semantic hashing

Auto-encodeur débruiteur



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 :

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

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 (cf. figure ci-dessous).

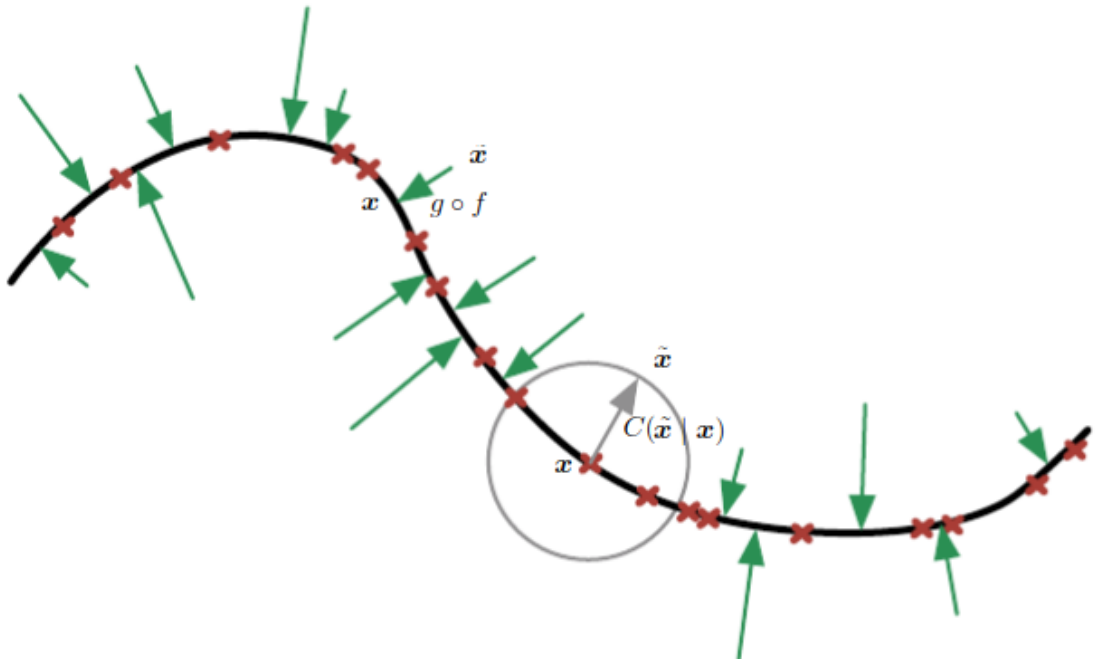


Image de [Goodfellow et al.] Deep Learning

Auto-encodeur débruiteur - quelques applications

Débruitage de signal sonore :

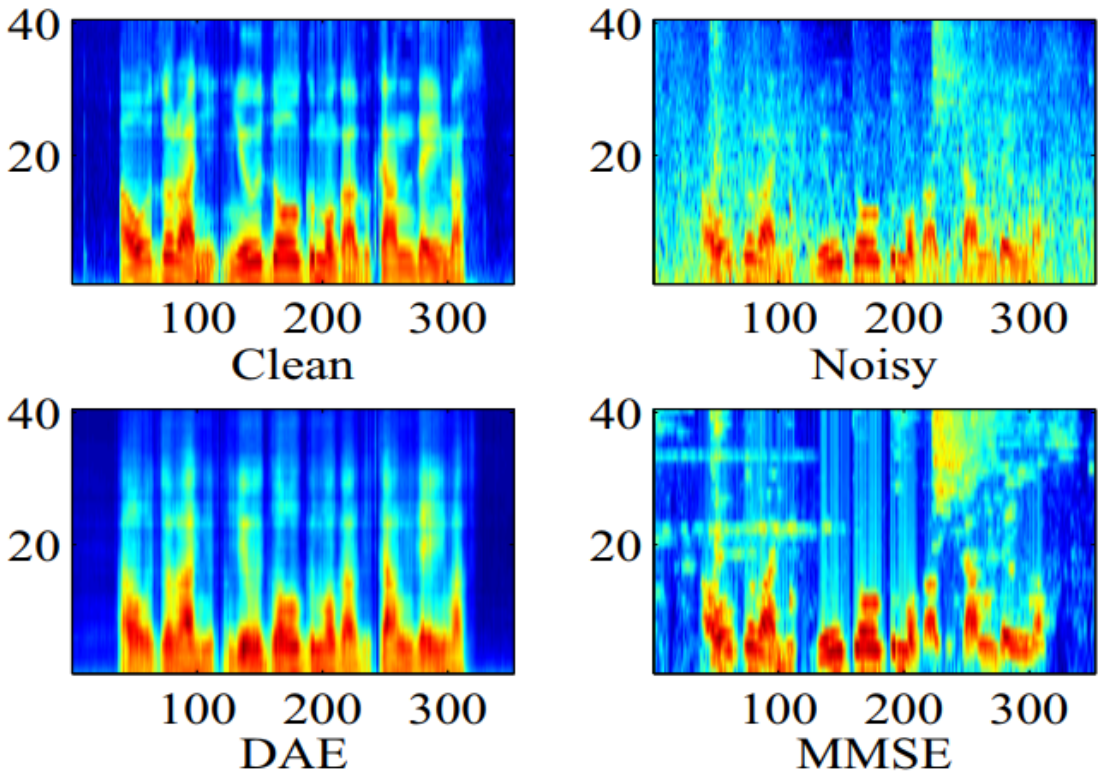


Image de [Lu et al.] Speech Enhancement Based on Deep Denoising Autoencoder

Auto-encodeur débruiteur - quelques applications

Inpainting :

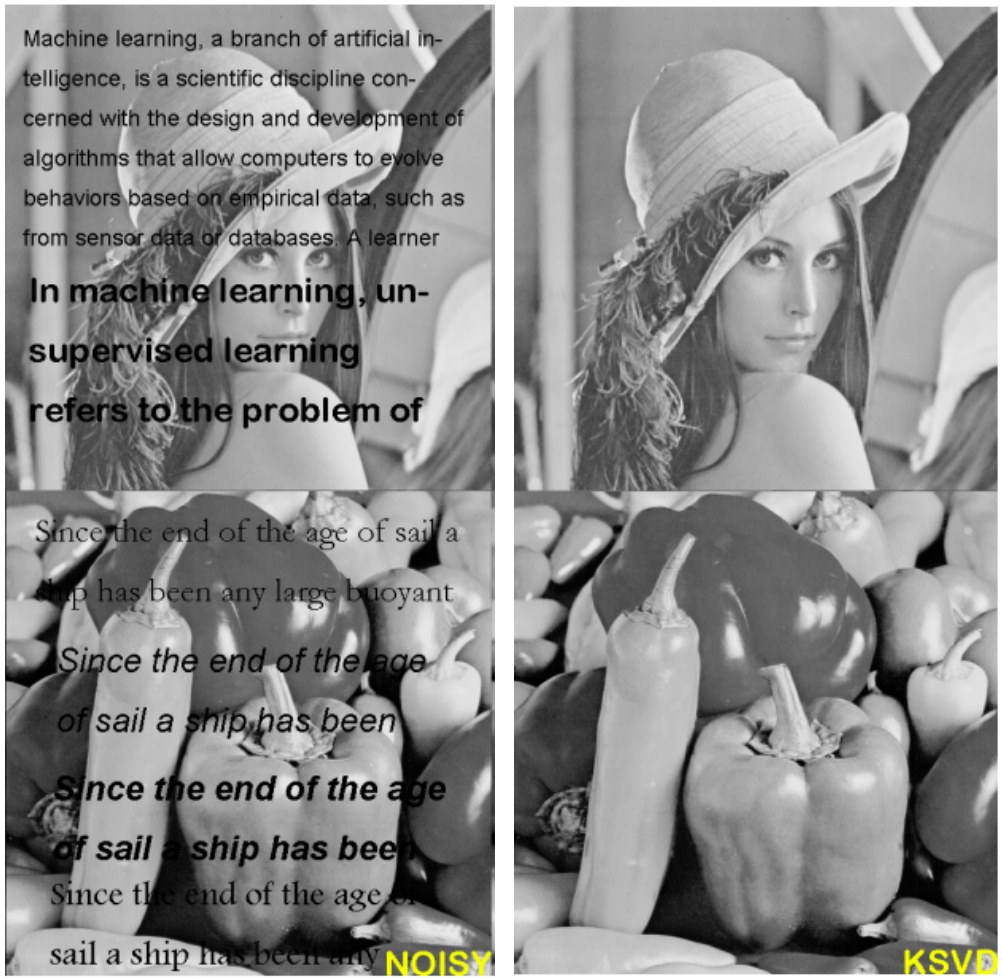
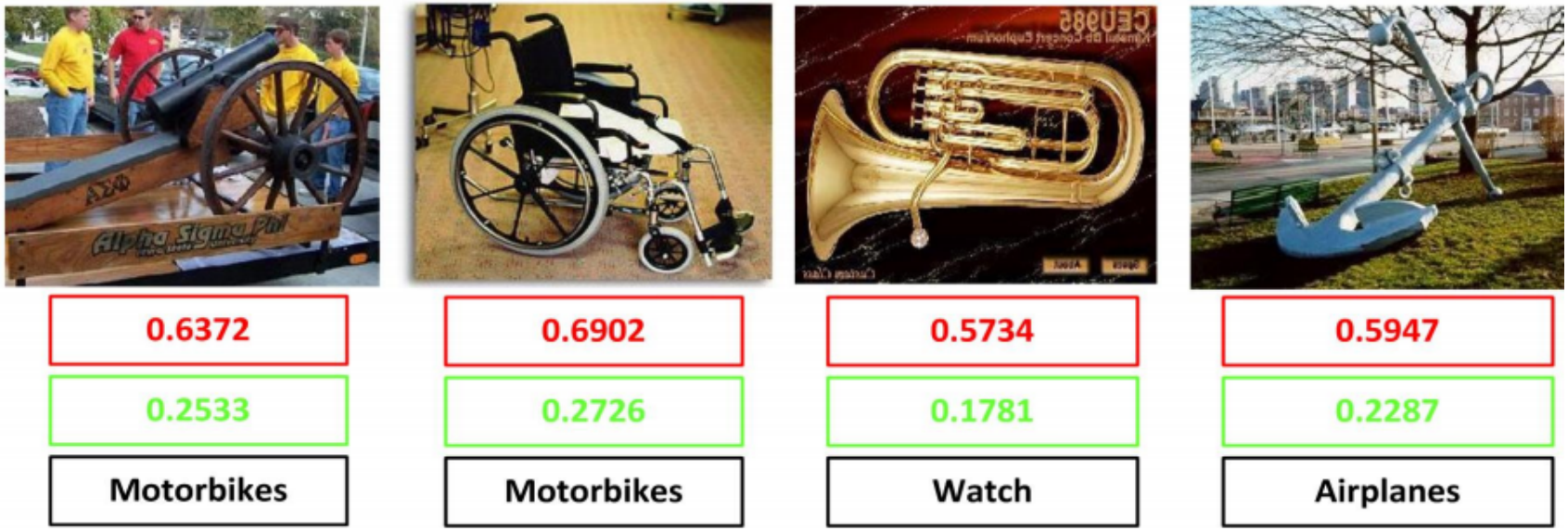


Image de [Xie et al.] Image Denoising and Inpainting with Deep Neural Networks

Auto-encodeur et détection d'anomalies

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.

Image de [Zhai et al.] Deep Structured Energy Based Models for Anomaly Detection

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...

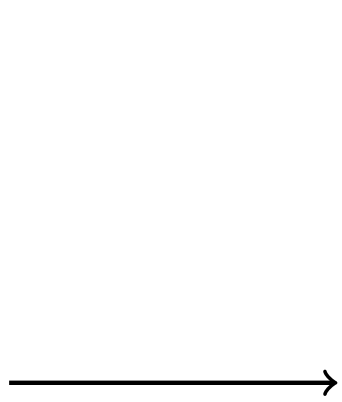
Modèles génératifs

... avec les modèles génératifs !

Qu'est-ce qu'un modèle génératif ?

Données d'entraînements

$p_{\text{données}}(x)$



Données générées

$p_{\text{modèle}}(x)$

Étant donné un ensemble de données dites d'entraînement (échantillons), nous voulons être capable de générer des nouvelles données suivant la même distribution.

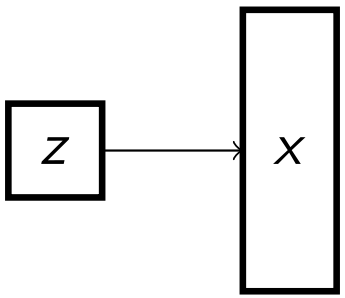
Modèles génératifs - Applications

Les applications de ces modèles génératifs sont très nombreuses :

- *Inpainting*
- Super-résolution
- Colorisation
- Art
- etc. (création de *fake news*, projection de votre visage dans 50 ans)
- Compression !

Modèles génératifs - Modélisation du problème

On commence par supposer que nos données x sont issues de variables latentes z. Par exemple, des photos de voitures représentent des véhicules d'une certaine marque, d'un certain modèle, d'une certaine couleur, prises sous un certain angle, etc.



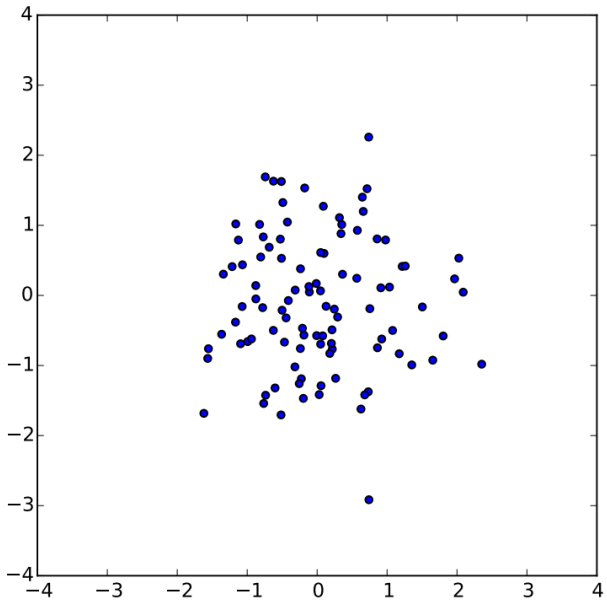
On cherche à apprendre une fonction qui génère les données x à partir de valeurs des variables latentes z.

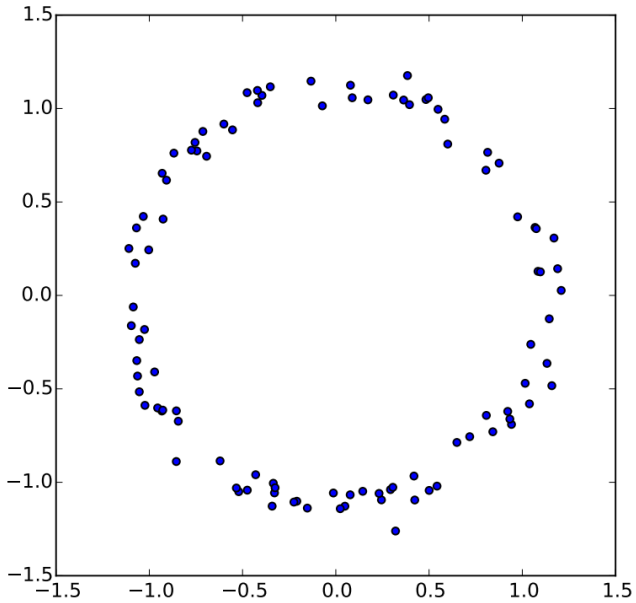
Modèles génératifs - Hypothèses

On peut (hypothèse classique) supposer que z suit une loi Gaussienne :

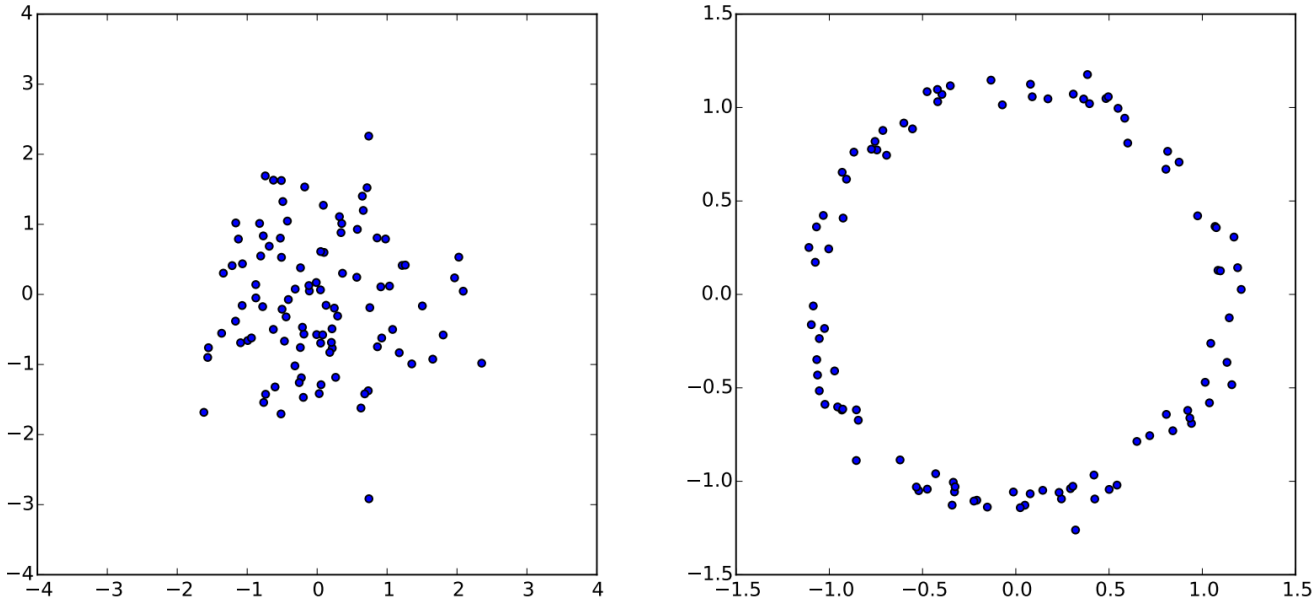
$$z \sim \mathcal{N}(0, I)$$

N.B. Choisir la matrice identité comme matrice de variance-covariance n'est pas anodin ; on suppose ici que nos variables latentes sont indépendantes.





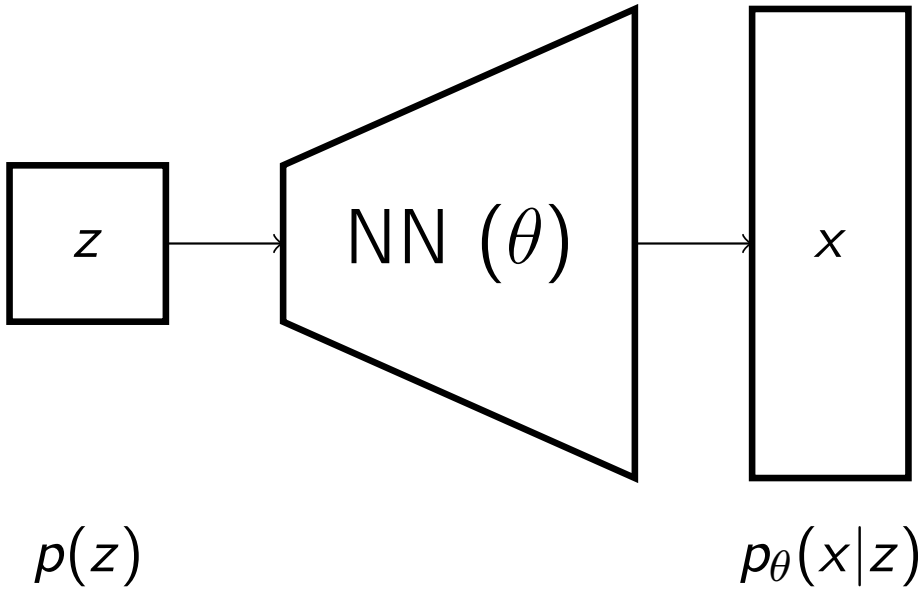
Modèles génératifs - Hypothèses



Il est possible de transférer une distribution gaussienne vers n'importe quelle distribution pour peu que l'on dispose d'un bon moyen d'approximer n'importe quelle fonction.

$\rightarrow$ On modélise cette fonction par un réseau de neurones !

Modèles génératifs - Entraînement



Pour entraîner ce réseau, on cherche à établir les paramètres θ qui maximisent la vraisemblance de générer nos données x.
Autrement dit, on veut résoudre :

$$\underset{\theta}{\text{argmax}}\, p_\theta(x)$$

Modèles génératifs - Entraînement

Or, on peut écrire :

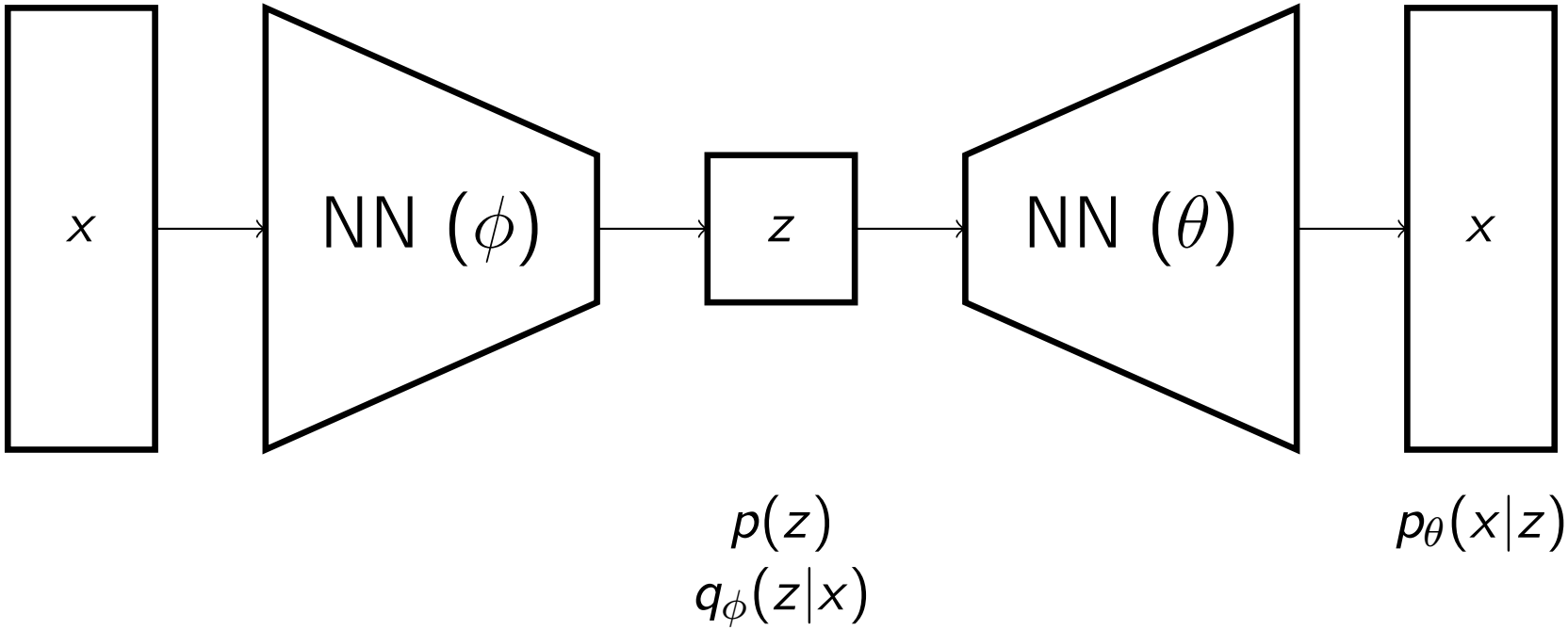
$$p_\theta(x) = \int_z p_\theta(z) p_\theta(x|z) dz$$

(formule des probabilités totales et théorème de Bayes)

Pour entraîner notre réseau de neurones génératif, il nous faut calculer cette intégrale sur z. **Impossible** !
(On pourrait l'estimer avec une méthode de Monte-Carlo, mais cela nécessiterait un trop grand nombre d'échantillons pour obtenir une estimation fiable à cause de la grande dimension de nos espaces).

Modèles génératifs - Entraînement

Pour résoudre ce problème, on introduit une fonction $q_\phi(z|x)$ qui définit une distribution des z qui ont une forte probabilité de générer les données x.



L'ensemble forme ce que l'on appelle un **auto-encodeur variationnel**.

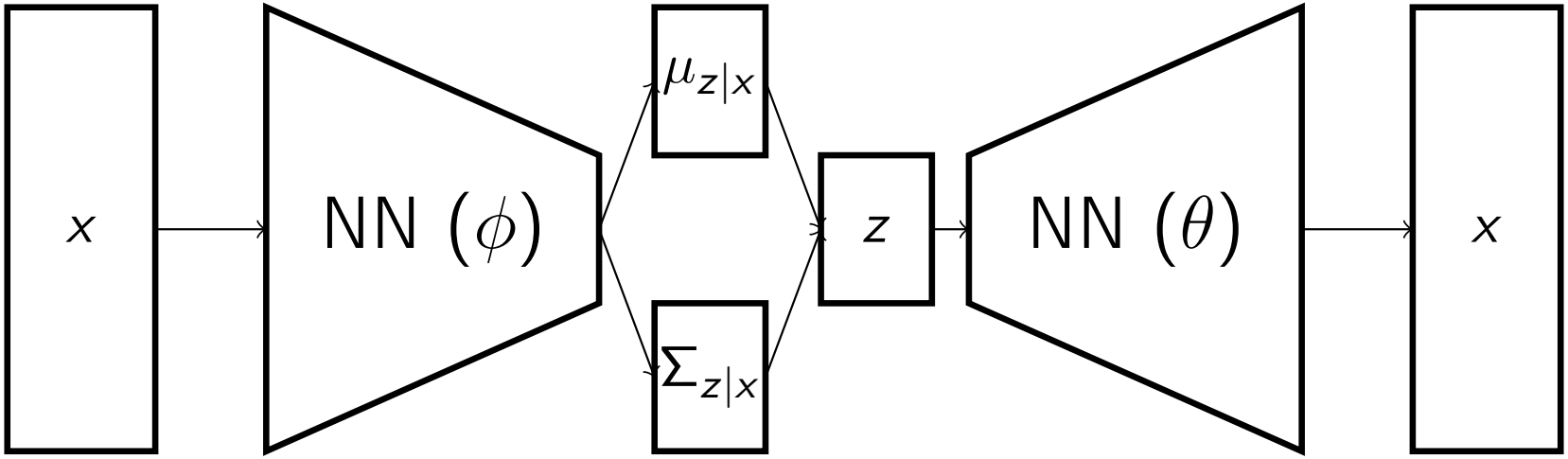
Auto-encodeurs variationnels

Une précision sémantique :

Les auto-encodeurs variationnels, en dépit de leur nom, sont assez différents des auto-encodeurs vus plus tôt dans le cours.
Ils partagent certes une architecture similaire et un vocabulaire commun (encodeur, décodeur, espace latent), mais les auto-encodeurs variationnels sont par nature **stochastiques**.

Les applications des auto-encodeurs s'appuient principalement sur la partie **encodeur** du réseau, alors que dans les auto-encodeurs variationnels, nous sommes intéressés par la partie **décodeur** qui constitue notre modèle génératif.

Auto-encodeurs variationnels - entraînement



Étant donnés un *minibatch* de données x, on génère leur représentation latente que l'on utilise pour estimer les paramètres de la distribution $z|x$: $\mu_{z|x}$ et $\Sigma_{z|x}$.
On échantillonne ensuite un z qui suit cette distribution et on l'utilise pour obtenir une donnée x.

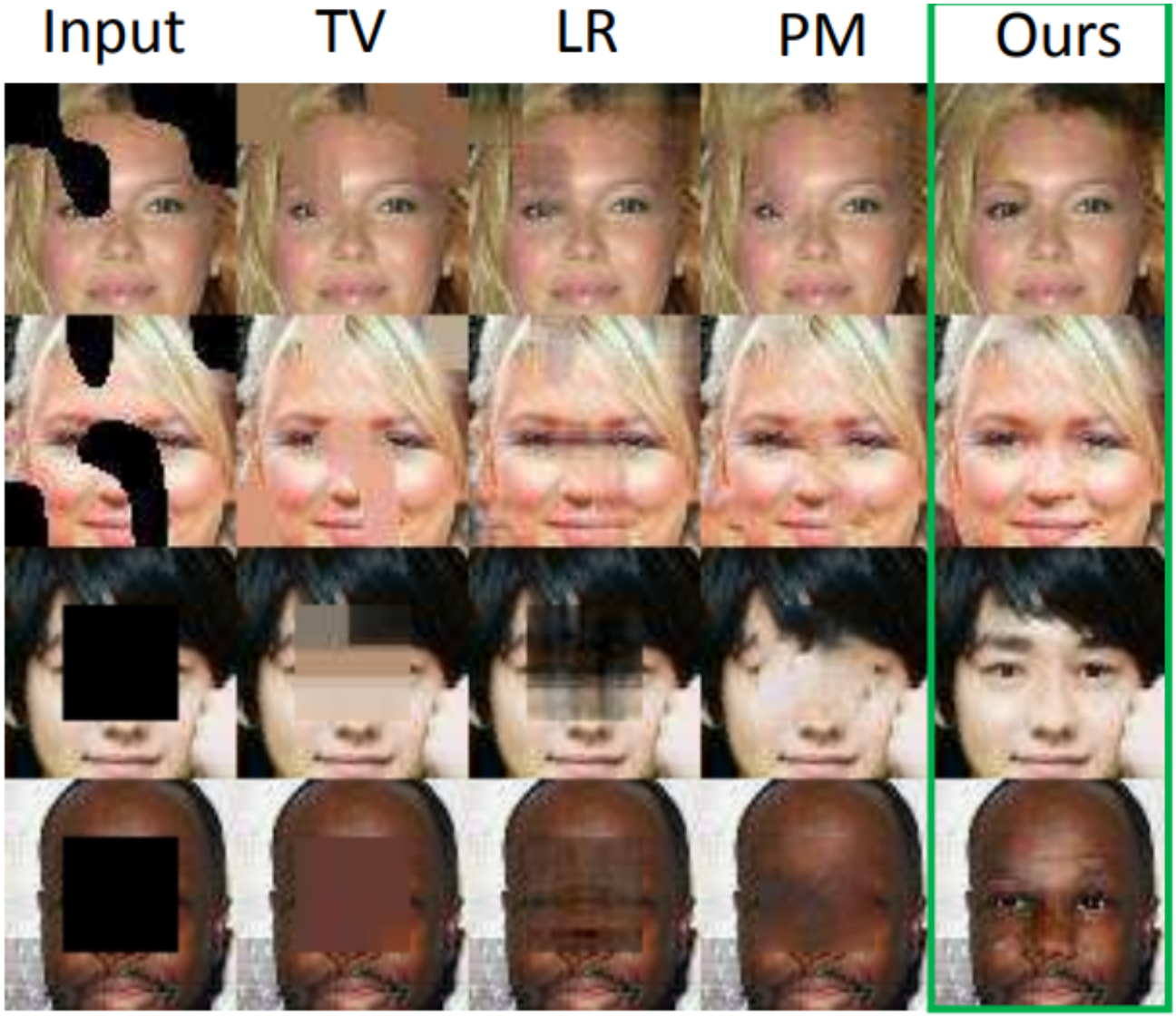
Auto-encodeurs variationnels

La fonctionnelle que l'on cherche à maximiser peut se réécrire dans ce contexte :

$$\underset{\theta}{\text{argmax}}\, \mathbb{E}_z[log(p_\theta(x|z))] - D_{KL}(q_\phi(z|x)||p_\theta(z)) \tag{1}$$

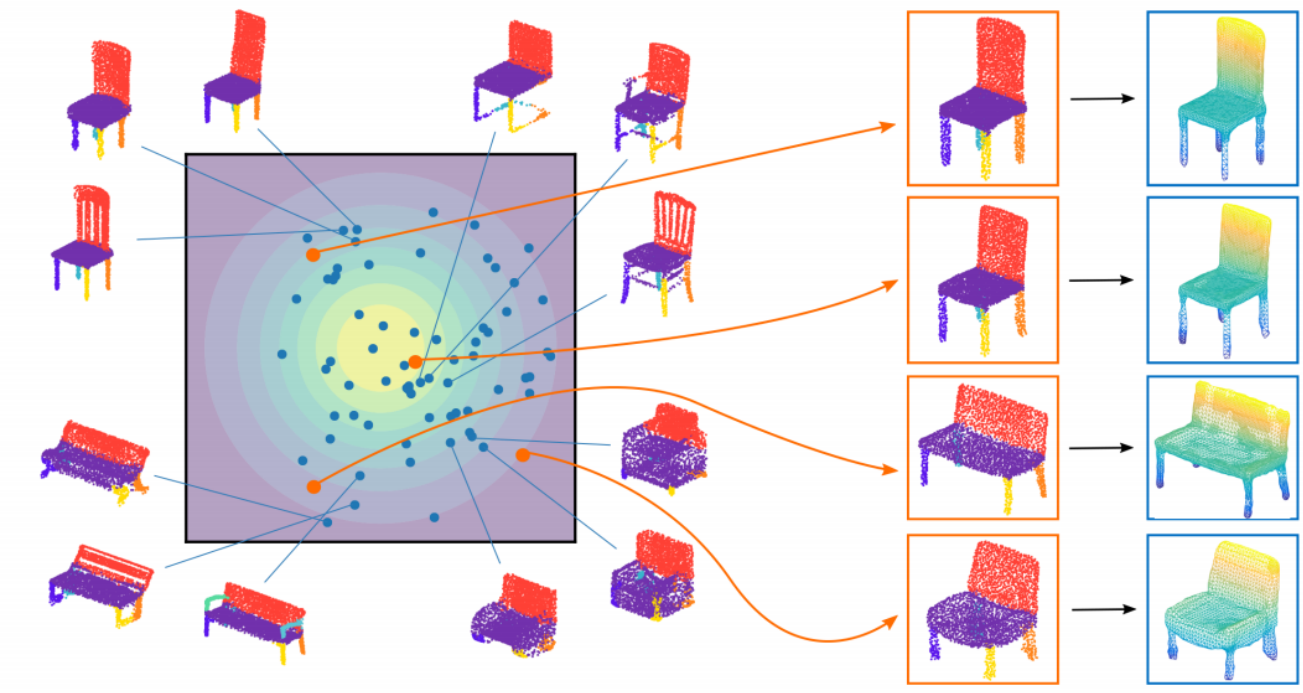
Le terme de gauche correspond à notre erreur de reconstruction, et le terme de droite est la divergence de Kullback-Leibler (une mesure de distance inter-distribution) entre notre *a priori* sur z et la distribution $z|x$ estimée par l'encodeur.
Par bonheur, cette expression est différentiable et nous pouvons entraîner le réseau de neurones par descente de gradient (moyennant une "astuce de reparamétrisation").

Auto-encodeurs variationnels - applications



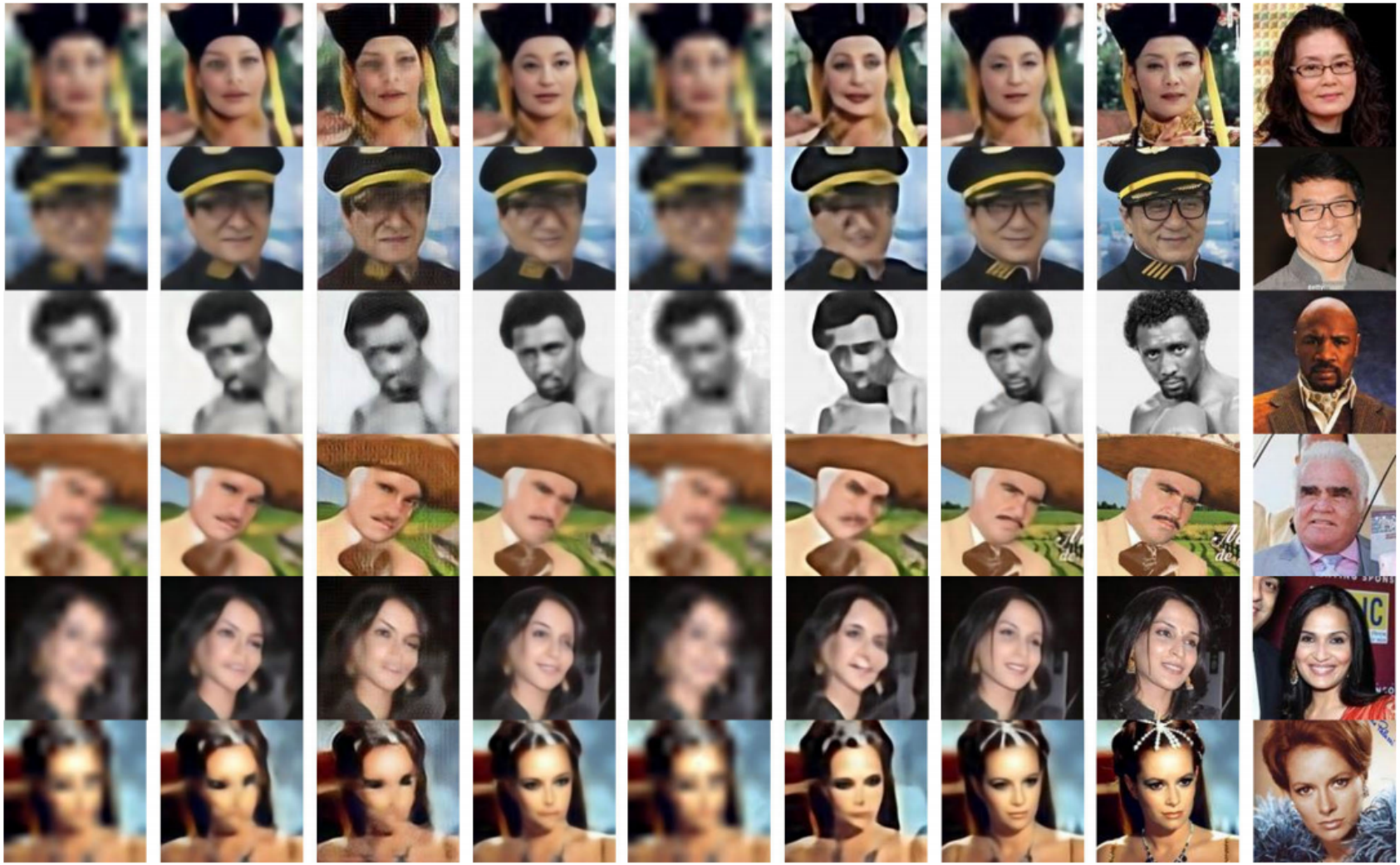
[Yeh et al.] Semantic Image Inpainting with Deep Generative Models

Auto-encodeurs variationnels - applications



[Nash et al.] The shape variational autoencoder : A deep generative model of part-segmented 3D objects

Auto-encodeurs variationnels - applications



[Liu et al.] Reference Based Face Super-Resolution

Suite du cours

- Modèles Génératifs Adversariaux (GAN)
- Apprentissage profond de données 3D
- 2e partie de l'APP !