Statistiques en grandes dimensions et Apprentissage profond :

Deep Clustering

2A ModIA

Contact :
Sandrine.Mouysset@irit.fr

"*Si l'intelligence était un gâteau, l'apprentissage non-supervisé serait le gâteau, l'apprentissage supervisé serait le glaçage et l'apprentissage par renforcement serait la cerise sur le gâteau.*"
Nous savons aujourd'hui faire le glaçage et la cerise, mais pas le gâteau.

Yann Le Cun (2016)

On ne dispose pas de base d'apprentissage/connaissance *a priori*/annotation, **l'apprentissage non supervisé** repose sur :

- définir des distances entre individus/variables ou des mesures de similarités,
- identifier des regroupements (agrégations/clusters).

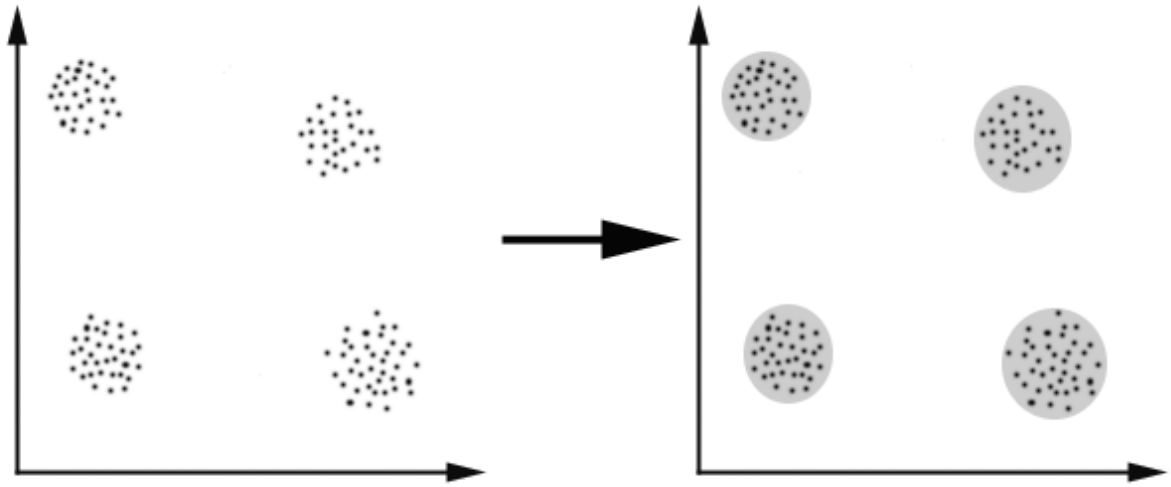


Figure: Exemple de classification non supervisée : diviser cet ensemble de points en 4 classes à partir de la distance entre les points

Il existe plusieurs méthodes de regroupement, basées alternativement sur :

- propriétés géométriques → K-means, Hiérarchique...
- propriétés spectrales → Classification spectrale...
- propriétés probabilistes → mélanges Gaussiens...

⇒ Classification non supervisée **en apprentissage profond** ?

Deep clustering

combinant l'extraction de caractéristiques, la réduction de la dimension et le clustering dans un modèle de bout en bout, permettant aux réseaux neuronaux profonds d'apprendre des représentations appropriées.

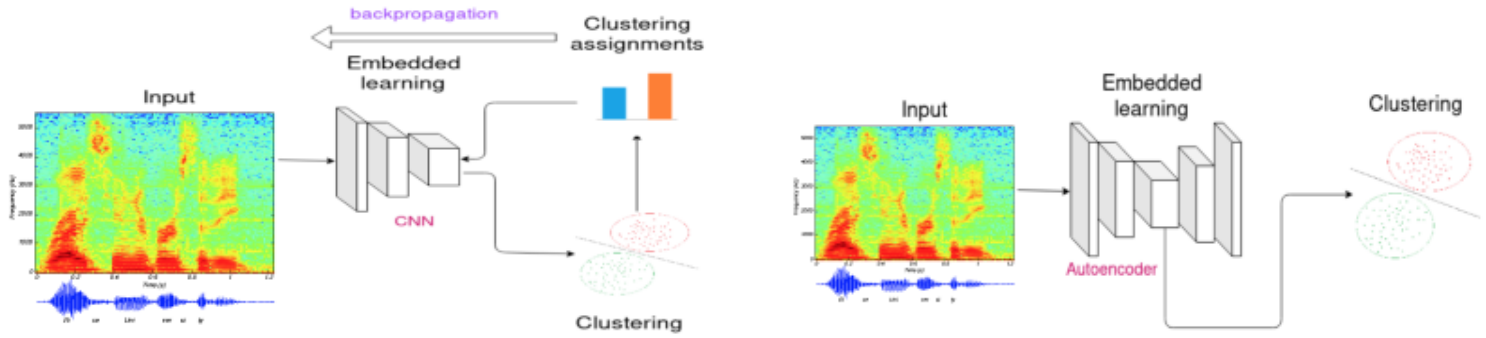


Figure: Exemple d'architectures Deep Clustering via CNN et Autoencodeur

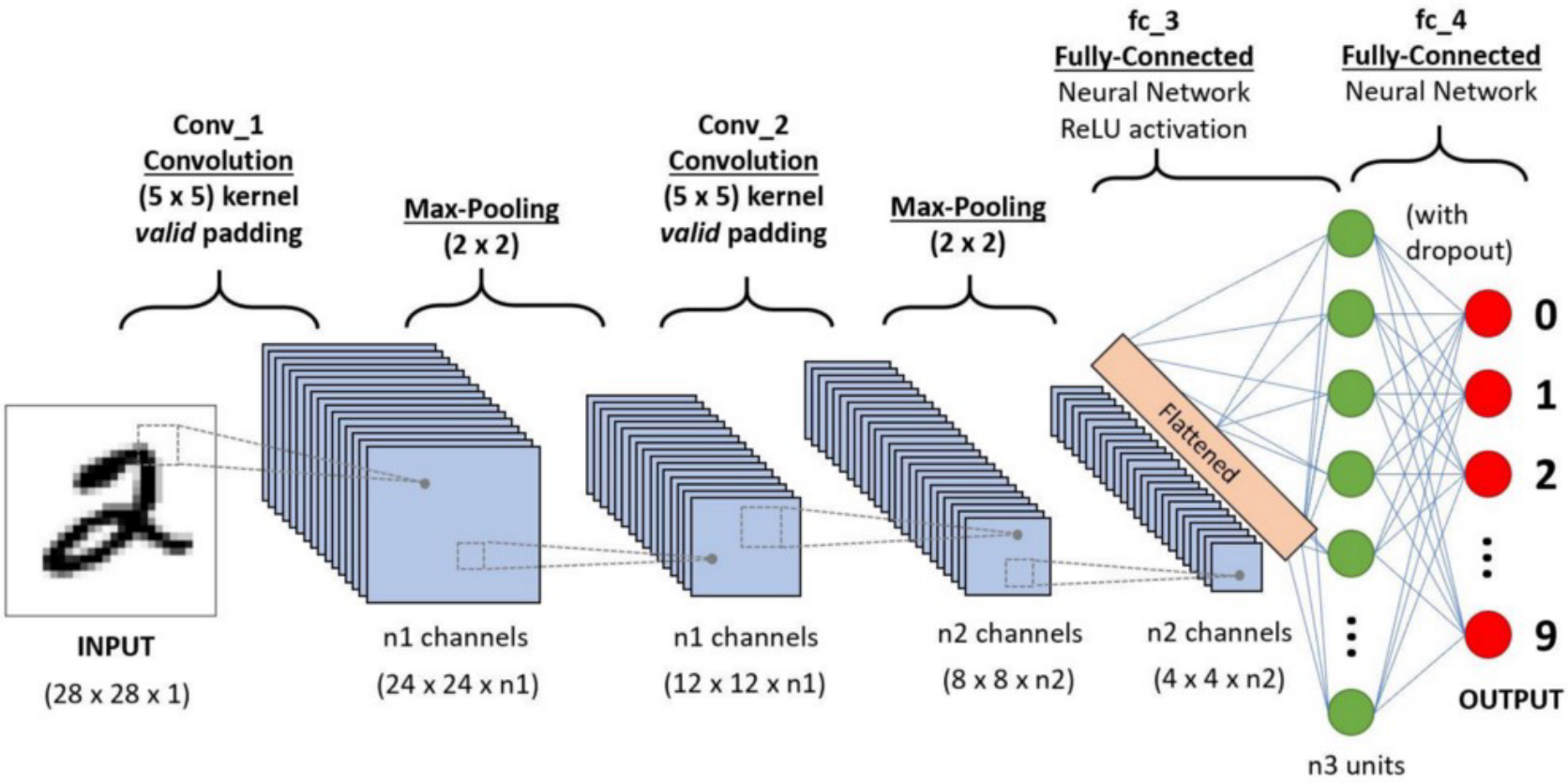


Figure: Architecture classique de CNN

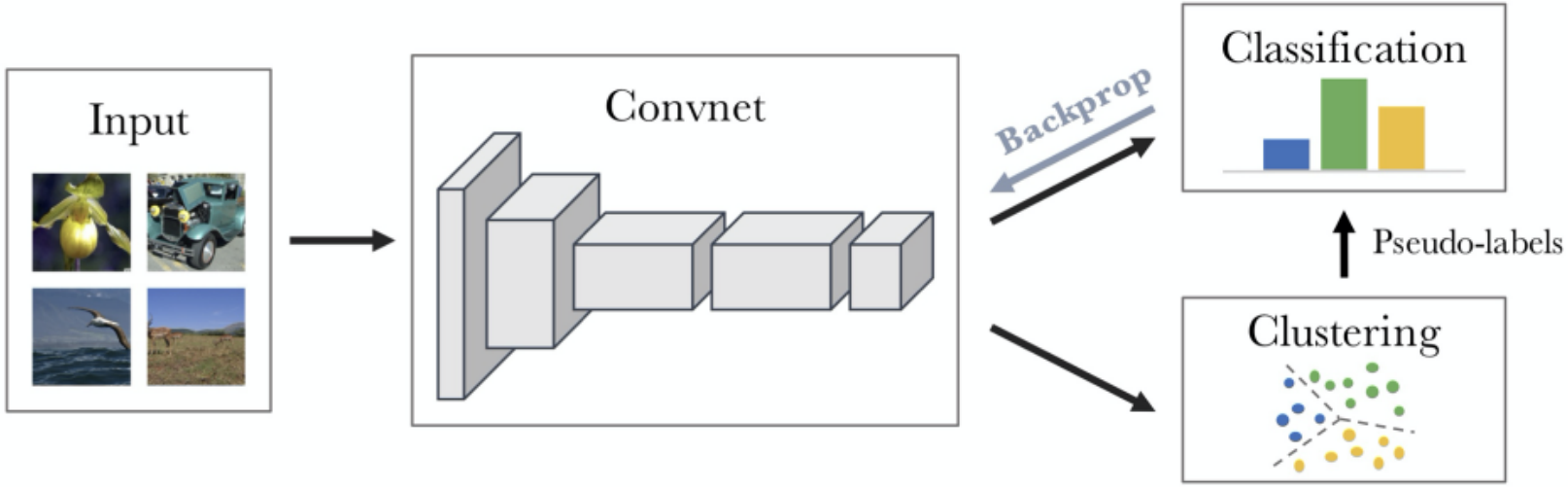


Figure: Architecture DeepCluster : regroupement itératif des caractéristiques par un algorithme de clustering standard, et utilisation des affectations ultérieures comme supervision pour mettre à jour les poids du réseau.

[M. Caron et al] Deep Clustering for Unsupervised Learning of Visual Features, 2018

DeepCluster alterne entre :

- le clustering des features pour produire des pseudo-étiquettes en utilisant :

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|f_\theta(x_n) - C y_n\|_2^2 \text{ tel que } y_n \mathbb{I}_k = 1$$

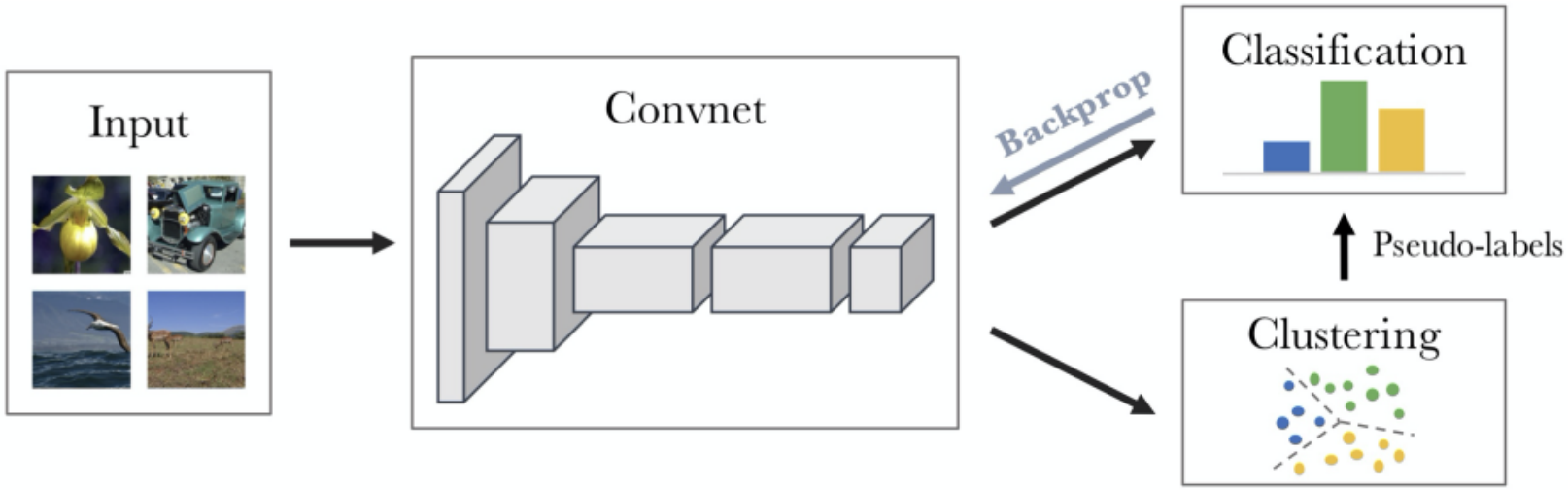
  avec C matrice des centres $d \times k$, $f_\theta(x_n)$ features produits par *convnet* et y_n la classe d'assignement de chaque image n

- et la mise à jour des paramètres du *convnet* en prédisant ces pseudo-étiquettes en utilisant :

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(g_W(f_\theta(x_n)), y_n)$$

  avec $\mathcal{L}$ est la perte logistique multinomiale, les paramètres W du classificateur et le paramètre θ.

[M. Caron et al] Deep Clustering for Unsupervised Learning of Visual Features, 2018



Choix de la méthode de clustering : kmeans, hiérarchique, classification spectrale, mélange gaussien...
$\Rightarrow$ paramètres inhérents à ces méthodes à "tuner", nombre de classes à définir, classes vides.

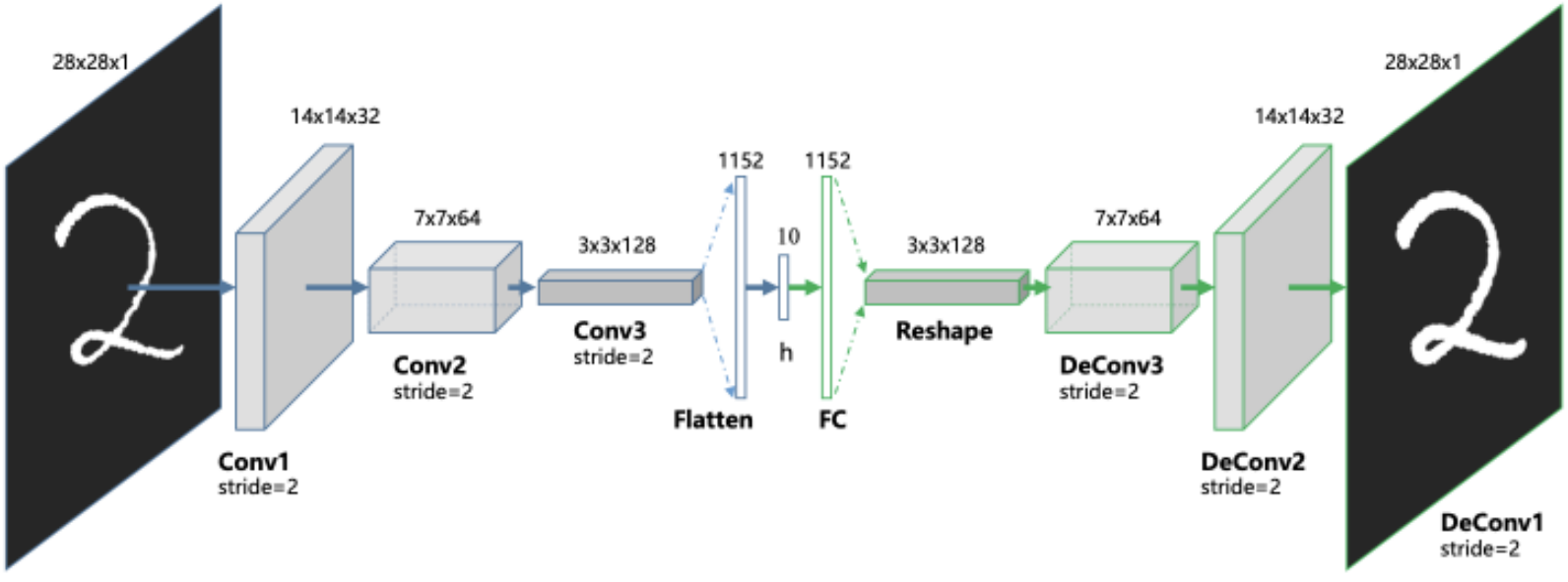


Figure: Architecture d'autoencodeur : algorithme d'apprentissage non supervisé par nature, car il ne prend en compte que les images elles-mêmes et n'a pas besoin d'étiquettes pendant la formation.

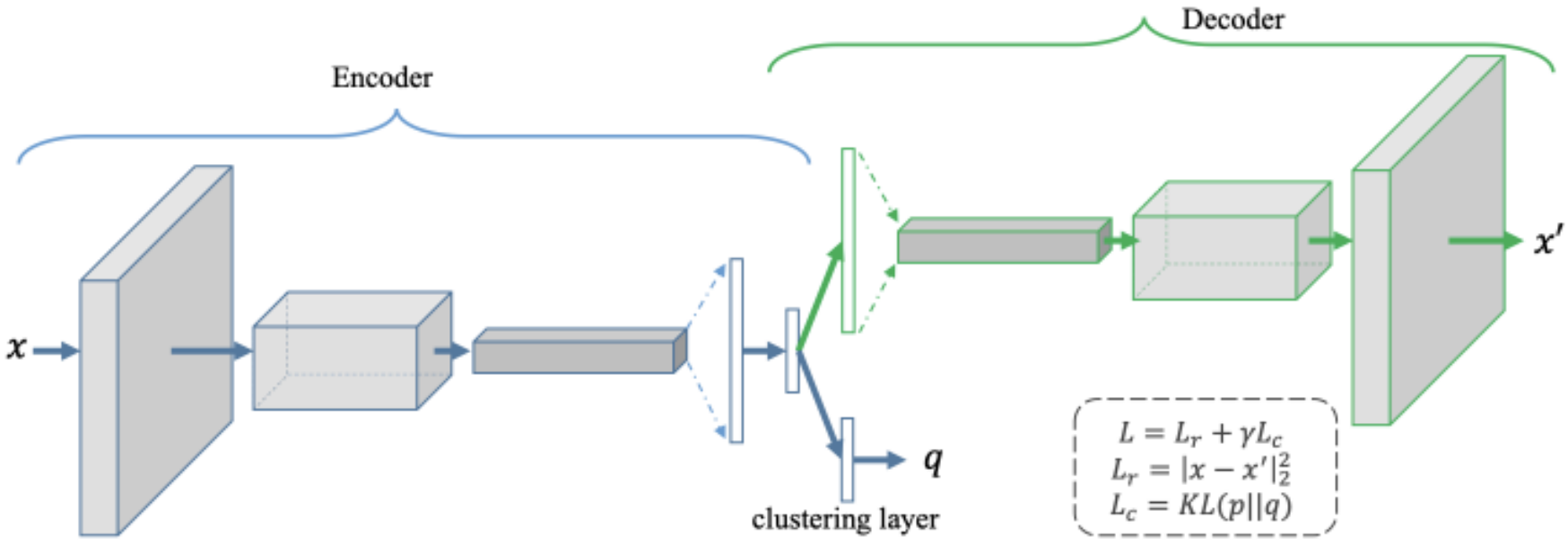


Figure: Architecture DCEC dont l'objectif est défini par $L = L_r + \gamma L_c$ où L_r la perte de reconstruction et L_c la perte de clustering, et $\gamma > 0$ est un coefficient qui contrôle le degré de distorsion de l'espace latent.

[X. Guo et al] Deep Clustering with Convolutional Autoencoders , 2018

Deep Convolutional Embedded Clustering (DCEC) comprend :

- La **couche de clustering** maintient les centres de cluster μ_j comme poids entraînables et projette chaque point intégré z_i en soft label q_i par la distribution t de Student (utilisée comme noyau pour mesurer la similarité entre $z_i = f_\theta(x_i)$ et le centroïd μ_j):

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_j (1 + \|z_i - \mu_j\|^2)^{-1}}$$

où q_{ij} est la $j$ième entrée de q_i, représentant la probabilité d'appartenance de z_i au cluster j.

- la **perte de clustering** L_c est définie par la divergence de Kullback-Leibler entre la distribution des soft labels et ceux prédéfinis par la distribution.

$$L_c = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

où P est la distribution cible définie comme

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}.$$

[X. Guo et al] Deep Clustering with Convolutional Autoencoders , 2018

La distribution cible a les propriétés suivantes :

1. renforcer les prédictions (c'est-à-dire améliorer la pureté des clusters),
2. mettre davantage l'accent sur les points de données attribués avec une confiance élevée,
3. normaliser la contribution à la perte de chaque centroïde pour éviter que les grands clusters ne déforment l'espace des caractéristiques cachées.

Fonction de perte :

$$L = L_r + \gamma L_c$$

Manipuler l'espace latent avec la perte de clustering L_c avec un paramètre, $\gamma < 1$ petit, préserve la propriété des autoencodeurs c'est-à-dire la structure locale de la distribution générant les données.

[X. Guo et al] Deep Clustering with Convolutional Autoencoders , 2018

Divergence de Kullback-Leibler (également appelée *entropie relative*) : Considérons deux distributions de probabilité P et Q : P représente les données, les observations, ou une distribution de probabilité mesurée et la distribution Q représente plutôt une théorie, un modèle ou une approximation de P.

La divergence de Kullback-Leibler mesure la distance statistique entre les 2 distributions de probabilité P et Q par :

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) log \frac{P(x)}{Q(x)}$$

Divergence de Kullback-Leibler (également appelée *entropie relative*) :
Considérons deux distributions de probabilité P et Q : P représente les données, les observations, ou une distribution de probabilité mesurée et la distribution Q représente plutôt une théorie, un modèle ou une approximation de P.

La divergence de Kullback-Leibler mesure la distance statistique entre les 2 distributions de probabilité P et Q par :

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) log \frac{P(x)}{Q(x)}$$

Interprétation :
La divergence de Kullback-Leibler s'interprète alors comme la différence moyenne du nombre de bits requis pour coder les échantillons de P en utilisant celui de Q.

Propriété : Quasi-distance !
(propriétés de symétrie et inégalité triangulaire non vérifiées)

La divergence de Kullback-Leibler mesure la distance statistique entre les 2 distributions de probabilité P et Q par :

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) log \frac{P(x)}{Q(x)}$$

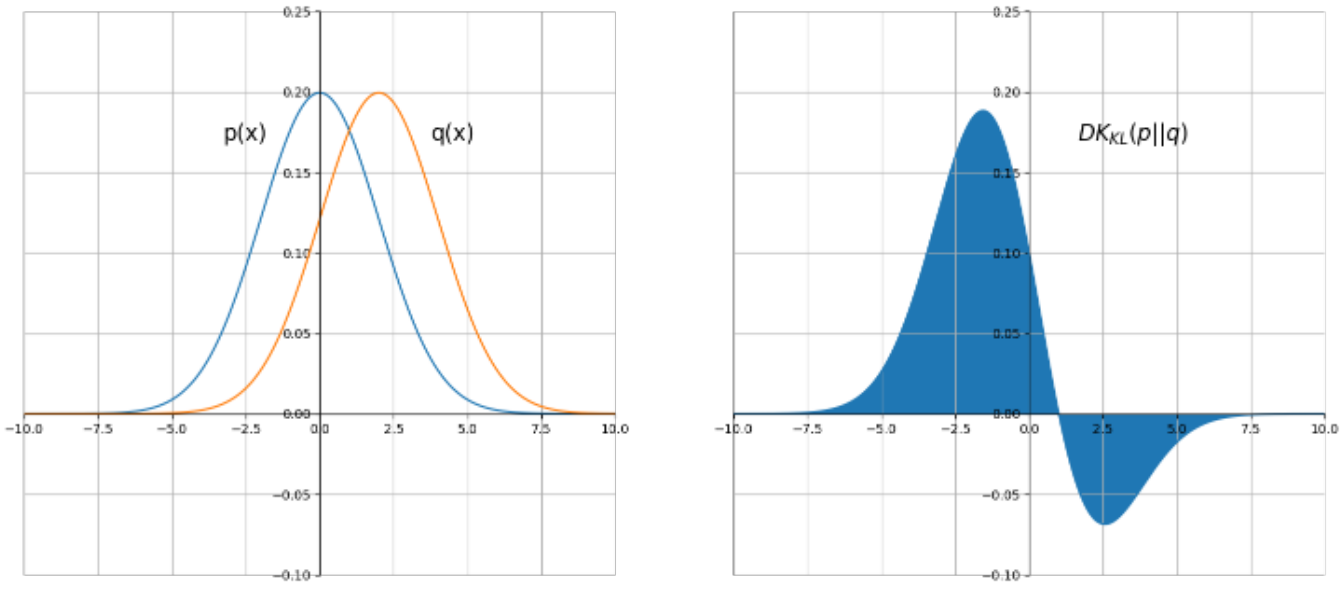


Figure: Exemple de divergence de Kullback-Leibler sur 2 distributions gaussiennes

Application : t-SNE (t-distributed stochastic neighbor embedding) méthode non linéaire permettant de représenter un ensemble de points d'un espace à grande dimension dans un espace 2D ou 3D en conservant la proximité entre les points pendant la transformation via la divergence KL.

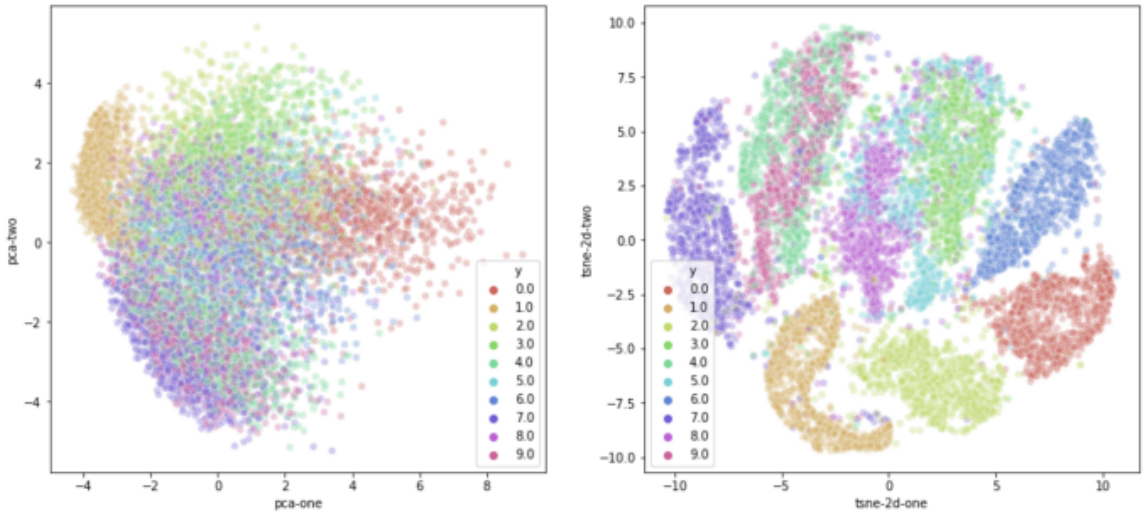


Figure: Exemple de réduction de dimensions par ACP et t-SNE

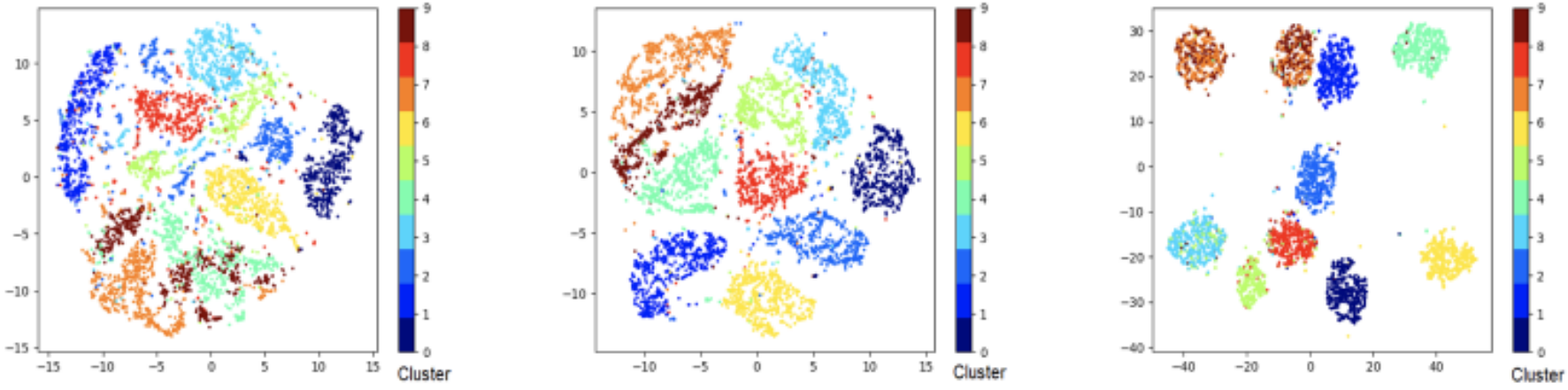


Figure: Exemple sur MNIST: (a) kmeans sur ensemble de données, (b) clustering sur espace latent de l'autoencodeur, (c) Deep Embedded Clustering

[E. Aljalbout et al] Clustering with Deep Learning: Taxonomy and New Methods, 2018

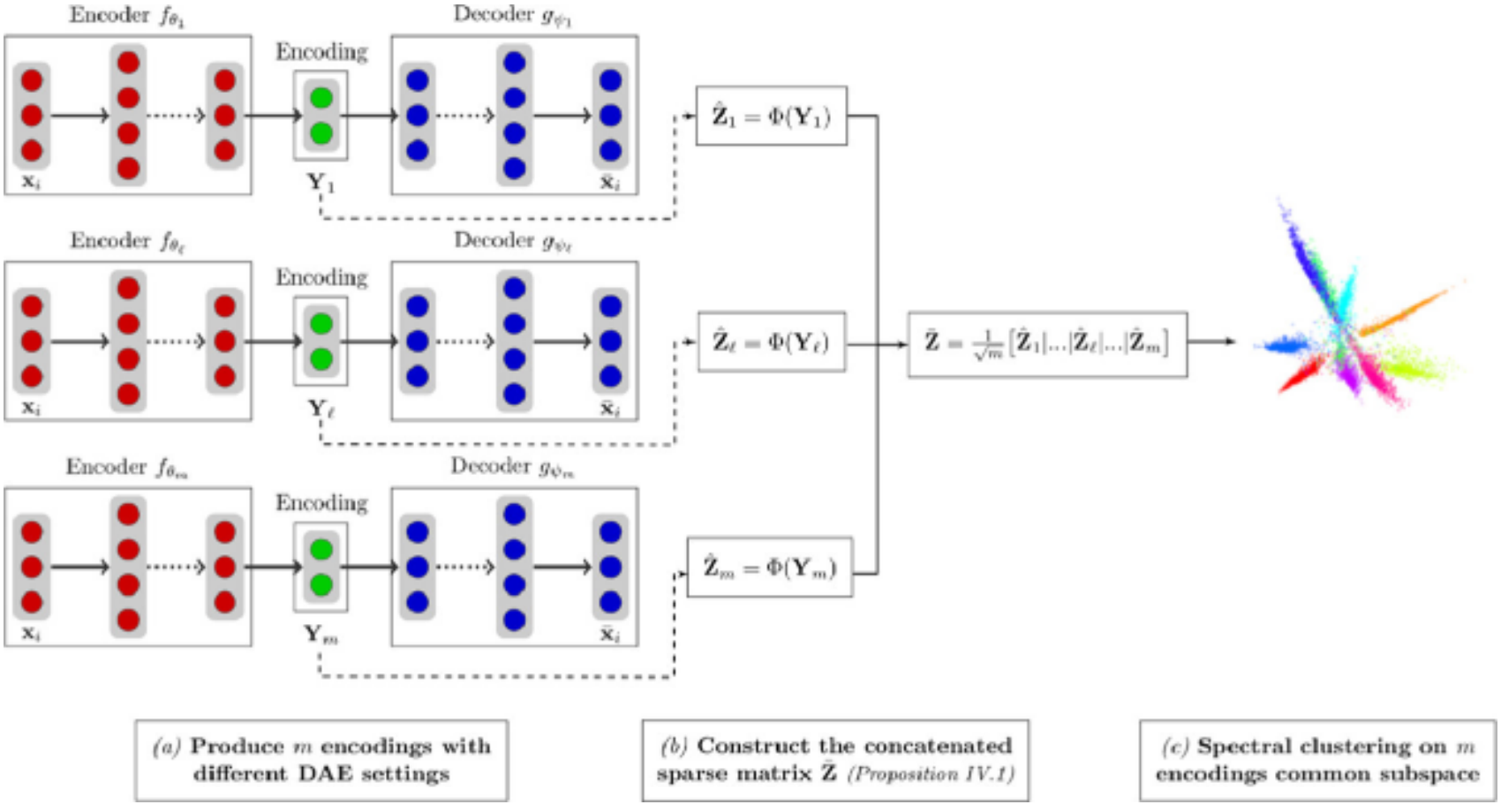


Figure: Autres méthodes de clustering dans l'espace latent des autoencodeurs : classification spectrale

[S. Affeldt et al] Spectral clustering via ensemble deep autoencoder learning, 2020

References

- GitHub sur Deep Clustering
  `https://github.com/zhoushengisnoob/DeepClustering`

- Librairie VISSL : A library for state-of-the-art self-supervised learning from images
  `https://vissl.ai`