



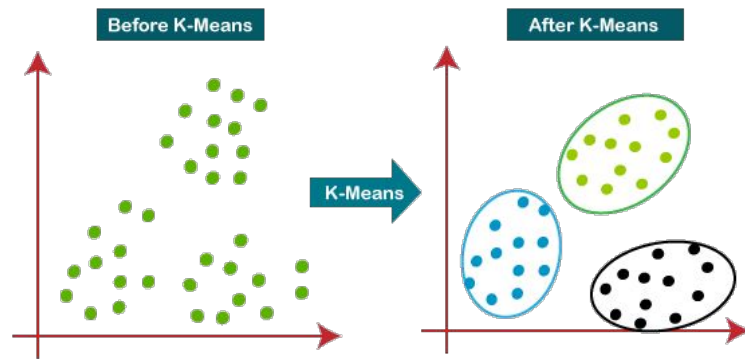
# Unsupervised Deep Learning

Legrand Thomas & Raphaël Pessis



# K-means : Principe

- Initialisation de k centroïdes
- .
- Calcul de la distance entre les points et tous les centroïdes
- Assignation des points au centroïde le plus proche.
- Calcul des nouveaux centroïdes en fonction de la moyenne des points assignés au cluster





# K-means : Méthodes

Visualisation :

- Visualisation des différents clusters formés avec les points qui leur appartiennent

Compression / Décompression :

- Remplacement de chaque point par le centroïde de son cluster

Génération :

- Utilisation d'un représentants de cluster
- Ajoute un bruit gaussien (normal) au centroid

# K-means : premiers résultats

Original Images

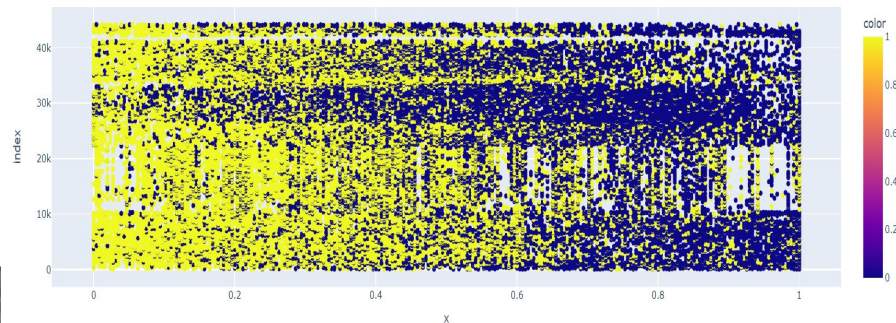


Nombre clusters	2
Itérations max	100
Tolérance	1e-3

Decompressed Images



Generated Images



# K-means : meilleurs résultats

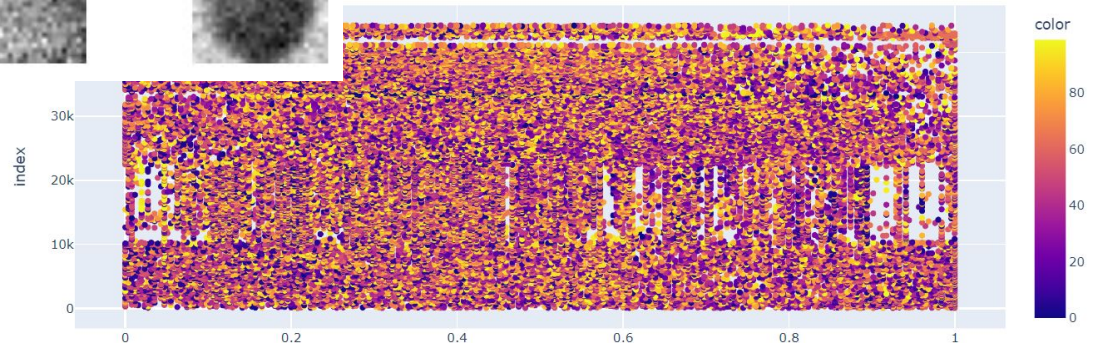
Decompressed Images



Generated Images



Nombre clusters	100
Itérations max	10000
Noise level	0.1





# PCA : Principe

- Normalisation des données
- Calcul des vecteurs et de leurs valeurs propres
  - Vecteurs : directions des axes de la variance de données
  - Valeurs : quantité de variance des vecteurs
- Sélection des k premiers vecteurs propres avec la plus grosse valeur propre



# PCA : Méthodes

Compression / Décompression :

- Projection des données sur K premier vecteurs propres
- Reconstruction des données sur les même vecteurs propres, application de la transposée

Visualisation 2D/3D :

- Projection des K premiers vecteurs propres sur un espace en k dimensions

Génération :

- Génération aléatoire à partir de la moyenne et de l'écart-type des composantes principales du dataset compressé

# PCA : premiers résultats

K

10

Image originale



Image originale



Image originale



Image originale

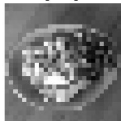


Image originale



Image décompressée

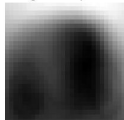


Image décompressée



Image décompressée

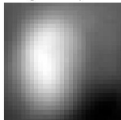


Image décompressée

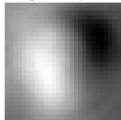
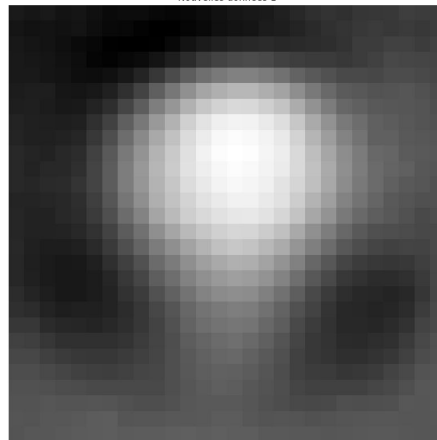


Image décompressée



Nouvelles données 1





# PCA : meilleurs résultats

K

500

Image originale

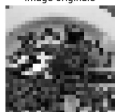


Image originale

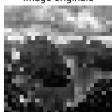


Image originale



Image originale

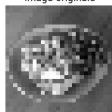


Image originale

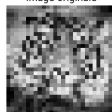


Image décompressée



Image décompressée

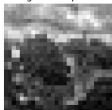


Image décompressée



Image décompressée

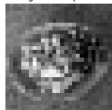
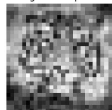
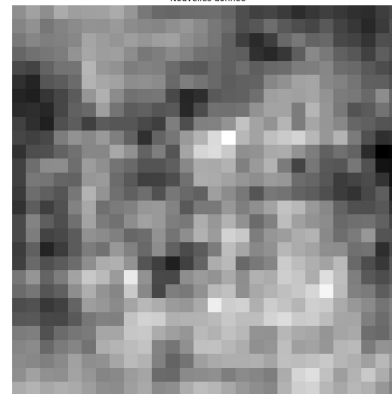


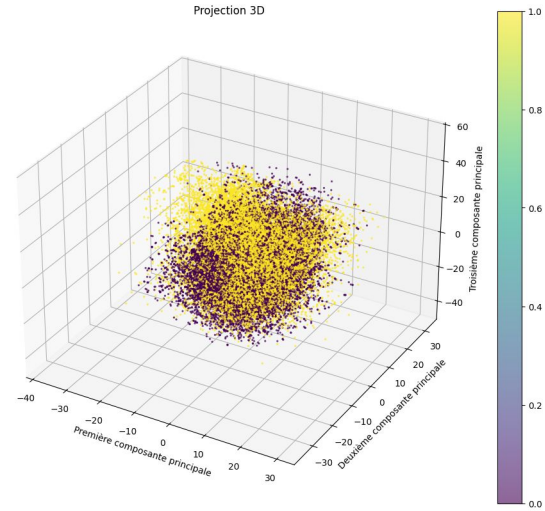
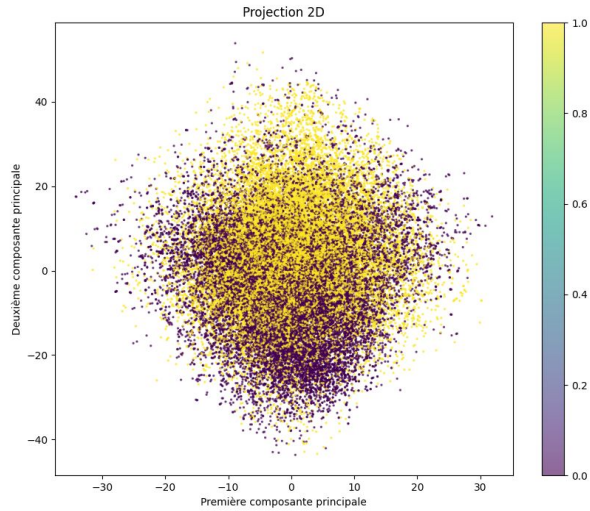
Image décompressée



Nouvelles donnée

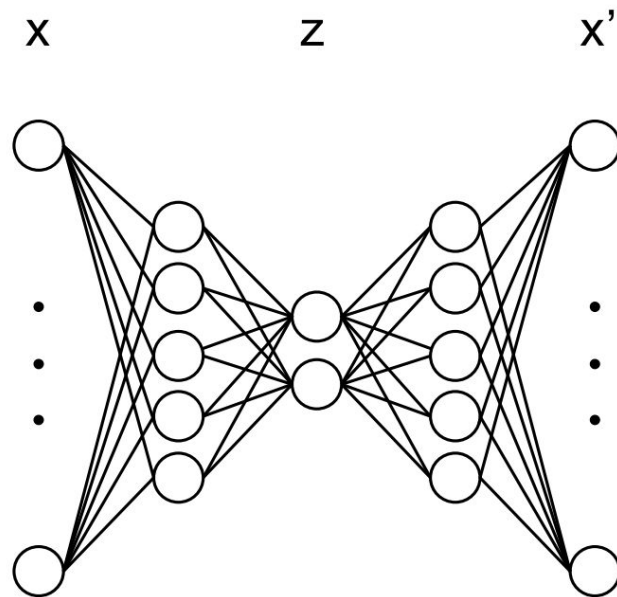


# PCA : visualisation



# Auto encoders : Principe

- Encodage des données par des couches Denses qui réduisent la dimension des données
- Reconstruction des données en appliquant les mêmes couches mais dans l'ordre inverse. On applique une sigmoïde pour la dernière couche pour que les valeurs des pixels restent entre 0 et 1



# Auto encoders : Méthodes



Visualisation :

- Projection de l'espace latent sur deux dimensions à l'aide des données compressées

Compression / Décompression :

- Utilisation des fonctions encodeur et décodeur (couches Dense)

Génération :

- Recherche aléatoire d'un vecteur latent avec distribution normale
- Utilisation du décodeur pour transformer le vecteur en image

# Auto encoders : premiers résultats



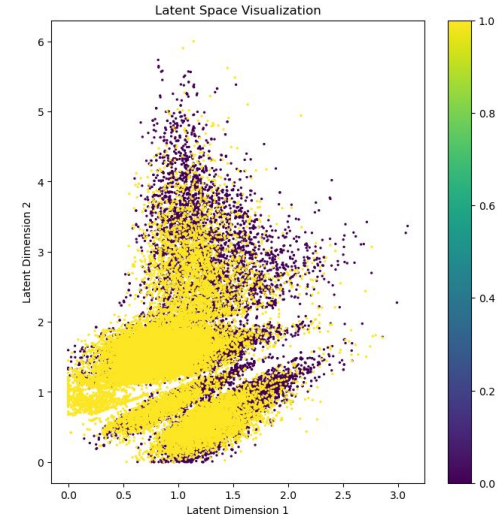
Decompressed Images



Generated Images



Encoding dimension	2
Epochs	100
Batch size	256
Fonction d'activation	relu
Fonction loss	mse
<u>noise_level</u>	0.1



# Auto encoders : meilleurs résultats

Original Images



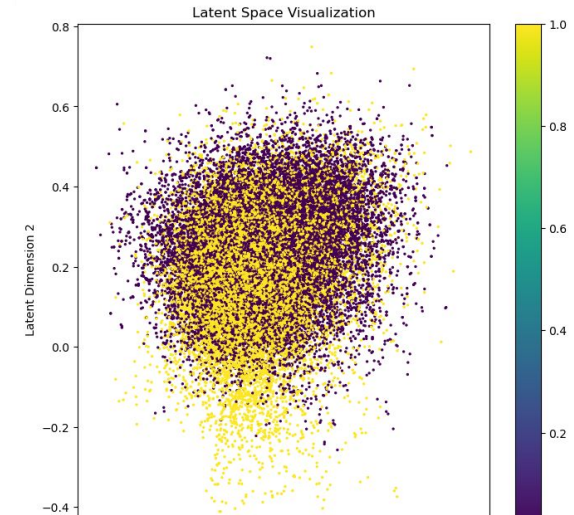
Decompressed Images



Generated Images



Encoding dimension	32
Epochs	500
Batch size	256
Fonction d'activation	tanh
Fonction loss	mse
Noise level	1



# SOMaps : Principe

- Création d'une grille 10x10 et initialisation du poids en random dans cette grille
- Calcul de la norme euclidienne entre chaque vecteur et les poids de chaque neurone (distance)
- Recherche du neurone avec le poids le plus proche du vecteur d'entrée
- Conversion de celui-ci en un indice 2D donnant les coordonnées du neurone dit "BMU"
- Modification du poids du BMU pour se rapprocher du vecteur d'entrée avec fonction de voisinage comprenant :
  - un taux d'apprentissage regressif
  - largeur de voisinage sigma adaptatif



# SOMaps : Méthodes

Visualisation :

Compression / Décompression :

- Extrait les poids des vecteurs présents dans `compressed_data` en fonction du BMU
- Crée un tableau numpy à partir de ces éléments où chaque ligne correspond au poids d'un neurone particulier

Génération :

- Sélection d'un neurone dans la grille SOM en random
- récupération de son poids
- Ajout du bruit et de l'écart-type

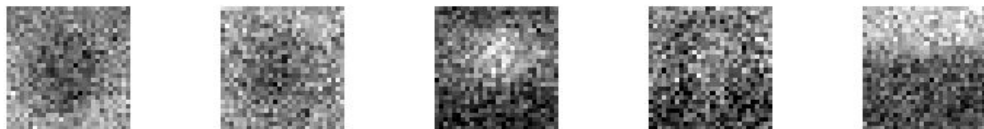


# SOMaps : premiers résultats

Decompressed Images



Generated Images



m	10
n	10
n_iterations	25
learning rate	0.5
noise_level (generation)	0.1

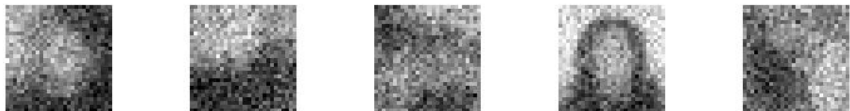


# SOMaps : meilleurs résultats

Decompressed Images



Generated Images

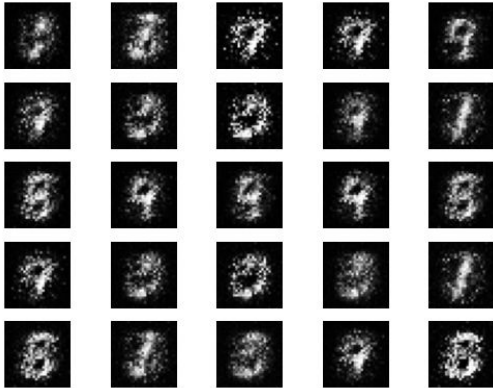


m	30
n	30
n_iterations	50
learning rate	0.7
noise_level (generation)	0.2



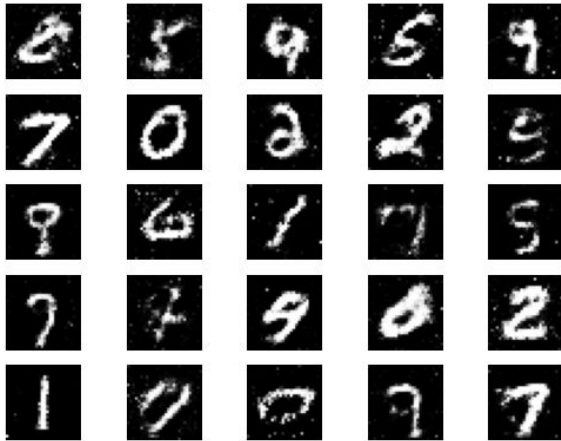
# GAN: Principe

# GAN : premiers résultats



Dimension Latente	10
Format de l'image	(28,28,1)
Nombre d'epoch	2000
Neurones générateur (Layer1, Layer2, Layer3)	128, 256, 512,
Neurones discriminator (Layer1, Layer2, Layer3)	256, 128, 1

## GAN : meilleurs résultats



Dimension Latente	100
Nombre d'epoch	9000
Intervalle d'Echantillonnage	1000
Neurones générateur (Layer1, Layer2, Layer3)	256, 512, 1024,
Neurones discriminator (Layer1, Layer2, Layer3)	512, 256, 1



## VAE : Principe

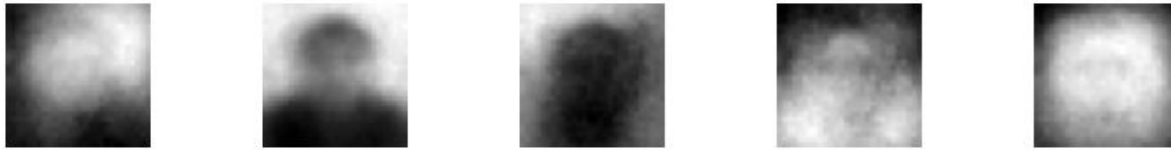
- Encodeur prend les données en entrée et cible une zone latente avec des paramètres de distribution
- Une couche lambda utilise ces paramètres pour créer des vecteurs latents
- Le décodeur reconstruit ces vecteurs latents en optimisant la perte de reconstruction et la perte de loss avec fonction KL

# VAE : premiers résultats

Original Images



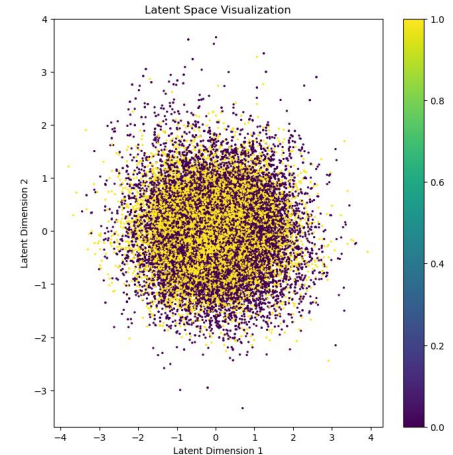
Decompressed Images



Generated Images



Encoding dimension	32
Epochs	300
Batch size	256
Fonction d'activation	tanh
Fonction loss	kl_loss perso
Noise level	0.5

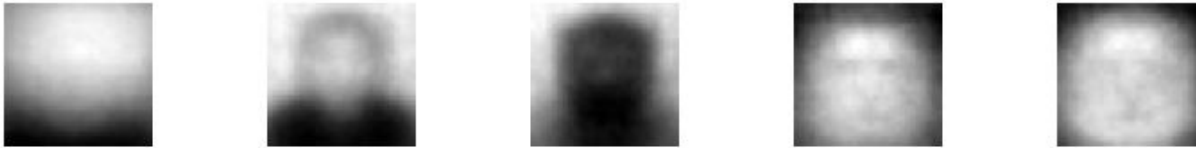


# VAE : meilleurs résultats

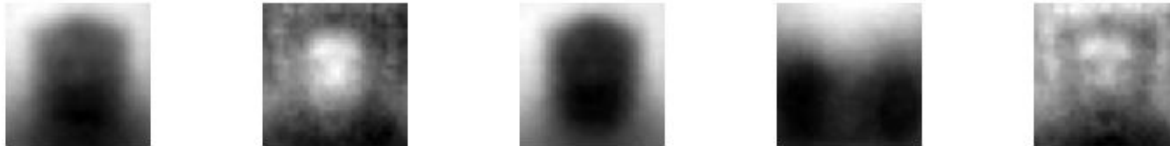
Original Images



Decompressed Images



Generated Images



Encoding dimension	2
Epochs	500
Batch size	256
Fonction d'activation	relu
Fonction loss	kl_loss perso
Noise level	0.5

