

# Auto-encodeur

Réduction de dimension et visualisation

Ricco Rakotomalala

Université Lumière Lyon 2



# Plan

1. Auto-encodeur - Principe
2. Pratique des auto-encodeurs
3. Traitement d'images
4. Conclusion
5. Références



Réseaux de neurones

# PRINCIPE DES AUTO-ENCODEURS



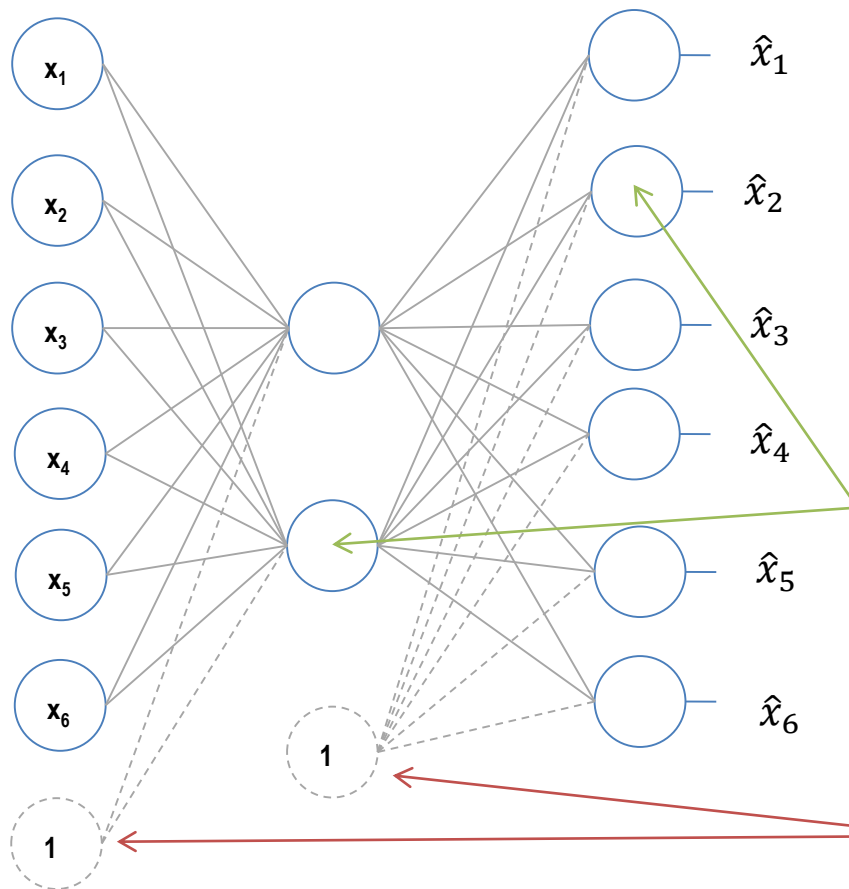
# Auto-encodeur ?

Dans sa forme la plus simple, un auto-encodeur est une sorte de [perceptron](#) à une couche cachée où les entrées et les sorties sont identiques. Nous sommes dans le cadre de l'apprentissage non-supervisé.

Critère possible à optimiser (MSE)

$$E = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p (x_{ij} - \hat{x}_{ij})^2$$

Et apprentissage des « poids synaptiques » par descente de gradient (cf. cours [Perceptron](#)).

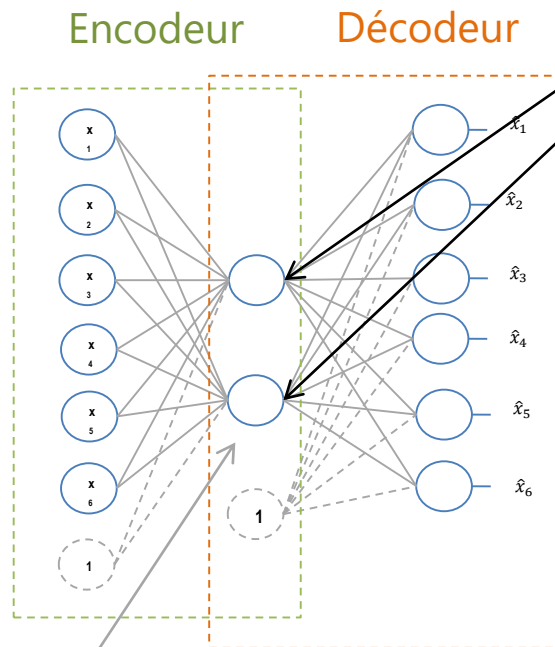


Possibilité d'utilisation différentes fonctions d'activation pour chaque couche.

Possibilité de rajouter un biais.



# Auto-encodeur – Quel intérêt ?



« **code** » qui permet de compresser (avec perte) l'information portée par les données, d'où le terme « auto-encodeur »

En sortie de la couche centrale, nous disposons d'une représentation des données dans un espace de dimension réduite.

Ex. Si fonction de transfert sigmoïde

$$v_1 = a_{1,0} + a_{1,1}x_1 + \dots + a_{1,6}x_6$$

$$v_2 = a_{2,0} + a_{2,1}x_1 + \dots + a_{2,6}x_6$$

$$u_1 = \frac{1}{1 + e^{-v_1}} \quad u_2 = \frac{1}{1 + e^{-v_2}}$$

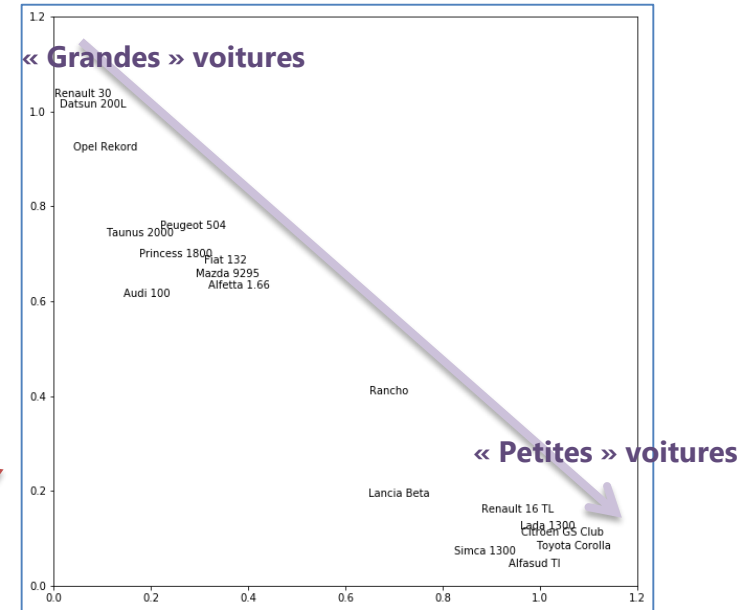
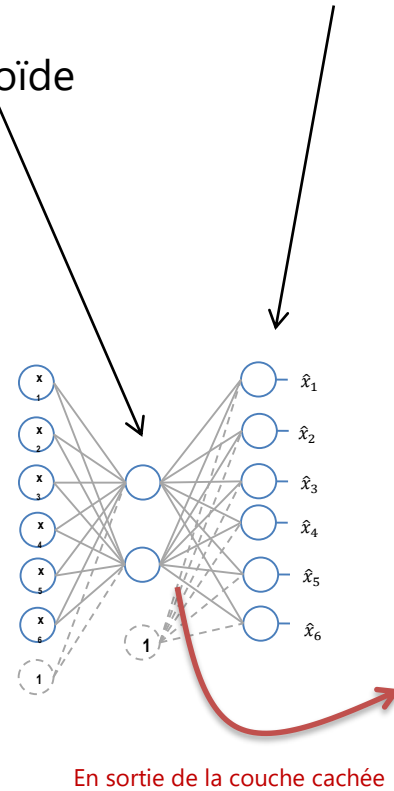
$(u_1, u_2)$  constitue un espace de représentation réduit préservant au mieux (au sens du critère MSE, d'autres critères sont possibles) les informations portées par les données : les « **pattern** ».



## Fonction de transfert linéaire

## Fonction de transfert sigmoïde

Modele	CYL	PUISS	LONG	LARG	POIDS	V.MAX
Alfasud TI	1350	79	393	161	870	165
Audi 100	1588	85	468	177	1110	160
Simca 1300	1294	68	424	168	1050	152
Citroen GS Club	1222	59	412	161	930	151
Fiat 132	1585	98	439	164	1105	165
Lancia Beta	1297	82	429	169	1080	160
Peugeot 504	1796	79	449	169	1160	154
Renault 16 TL	1565	55	424	163	1010	140
Renault 30	2664	128	452	173	1320	180
Toyota Corolla	1166	55	399	157	815	140
Alfetta 1.66	1570	109	428	162	1060	175
Princess 1800	1798	82	445	172	1160	158
Datsun 200L	1998	115	469	169	1370	160
Taunus 2000	1993	98	438	170	1080	167
Rancho	1442	80	431	166	1129	144
Mazda 9295	1769	83	440	165	1095	165
Opel Rekord	1979	100	459	173	1120	173
Lada 1300	1294	68	404	161	955	140

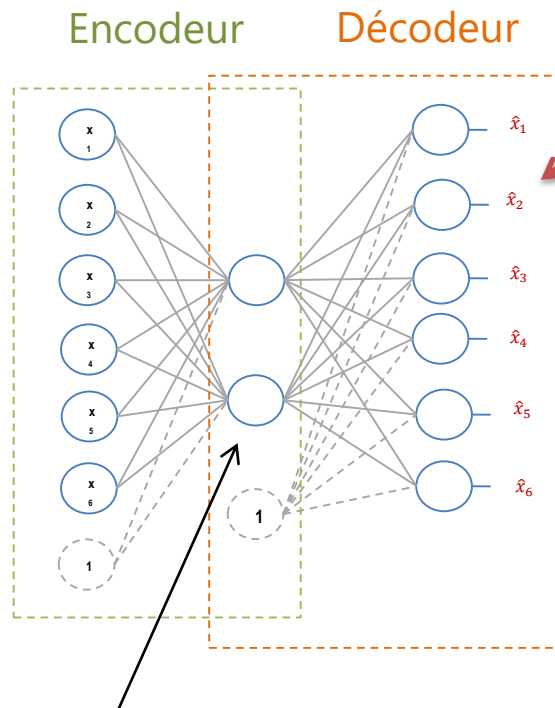


(Données « Autos » -- Saporta, 2006 ; page 428)

Dans cette configuration (une seule couche cachée sigmoïde), la solution produite par l'auto-encodeur est apparentée à celle d'une analyse en composantes principales.



# Auto-encodeur – Quel intérêt ? (bis)



En output de la couche de sortie, nous disposons d'une représentation dans l'espace initial expurgée du « bruit » et autres perturbations contenues dans les données.

Ex. Si fonction de transfert linéaire

$$\hat{x}_j = b_{j,0} + b_{j,1}u_1 + b_{j,2}u_2$$

On a une forme de nettoyage des données (filtrage) où seules les informations « utiles » (basées sur les « pattern ») sont conservées.

Si forts écarts, alors peut-être problèmes (points atypiques, ...) ou caractéristiques particulières, spécifiques.

Agit comme un goulot d'étranglement permettant de filtrer le signal « bruité » contenu dans les données.



## Données « Autos »

Mises en évidence ici les différences supérieures à (1.1 x écart-type)

Données initiales

Modele	CYL	PUISS	LONG	LARG	POIDS	V.MAX
Alfasud TI	1350	79	393	161	870	165
Alfetta 1.66	1570	109	428	162	1060	175
Audi 100	1588	85	468	177	1110	160
Citroen GS Club	1222	59	412	161	930	151
Datsun 200L	1998	115	469	169	1370	160
Fiat 132	1585	98	439	164	1105	165
Lada 1300	1294	68	404	161	955	140
Lancia Beta	1297	82	429	169	1080	160
Mazda 9295	1769	83	440	165	1095	165
Opel Rekord	1979	100	459	173	1120	173
Peugeot 504	1796	79	449	169	1160	154
Princess 1800	1798	82	445	172	1160	158
Rancho	1442	80	431	166	1129	144
Renault 16 TL	1565	55	424	163	1010	140
Renault 30	2664	128	452	173	1320	180
Simca 1300	1294	68	424	168	1050	152
Taunus 2000	1993	98	438	170	1080	167
Toyota Corolla	1166	55	399	157	815	140

Données reconstituées

Modele	CYL	PUISS	LONG	LARG	POIDS	V.MAX
Alfasud TI	1336	67	423	163	948	151
Alfetta 1.66	1745	91	437	168	1124	160
Audi 100	1734	91	439	168	1140	162
Citroen GS Club	1334	67	422	163	945	151
Datsun 200L	2000	105	445	170	1225	165
Fiat 132	1779	92	438	168	1134	160
Lada 1300	1340	67	422	163	947	151
Lancia Beta	1450	74	428	165	1009	155
Mazda 9295	1748	91	438	168	1127	160
Opel Rekord	1922	101	444	170	1201	164
Peugeot 504	1816	95	440	168	1154	162
Princess 1800	1781	93	439	168	1149	162
Rancho	1558	80	430	165	1038	156
Renault 16 TL	1387	70	424	164	968	152
Renault 30	2003	105	446	170	1227	165
Simca 1300	1380	70	425	164	972	153
Taunus 2000	1833	96	441	169	1170	163
Toyota Corolla	1322	66	422	163	938	151

*Eu égard à ses dimensions (plutôt bien approximées), la Renault 30 possède un gros moteur, elle est de ce fait puissante et rapide (par rapport à ce qu'on pouvait attendre)*

Version « épurée » des données traduisant les caractéristiques communes aux véhicules.

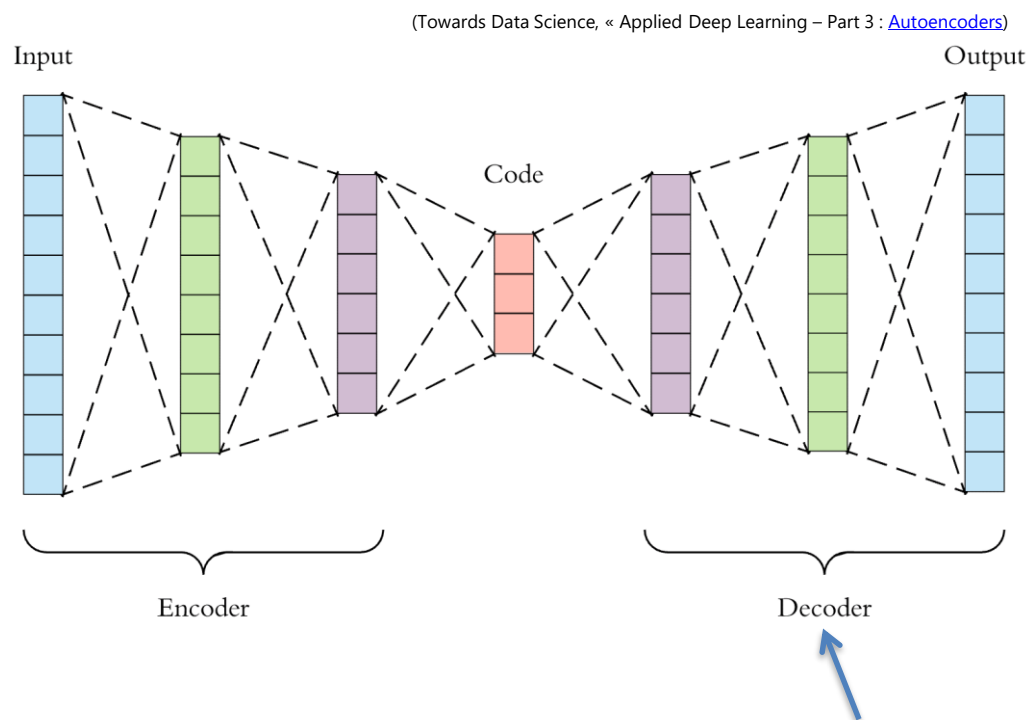
*Eu égard à leurs caractéristiques (cylindrée notamment), ces voitures italiennes des années 70 sont rapides.*





# Plus loin avec les auto-encodeurs

Il est possible d'ajouter des couches pour appréhender différents niveaux d'abstractions des données. Permet également de capter les « pattern » non-linéaires régissant les données.



Habituellement, nous plaçons des couches aux caractéristiques miroirs de la partie encodeur (ex. « Autos » : [6, 3, 2, 3, 6]).



Individus illustratifs, interprétation, variables illustratives

# PRATIQUE DES AUTO-ENCODEURS



# Positionner les individus supplémentaires

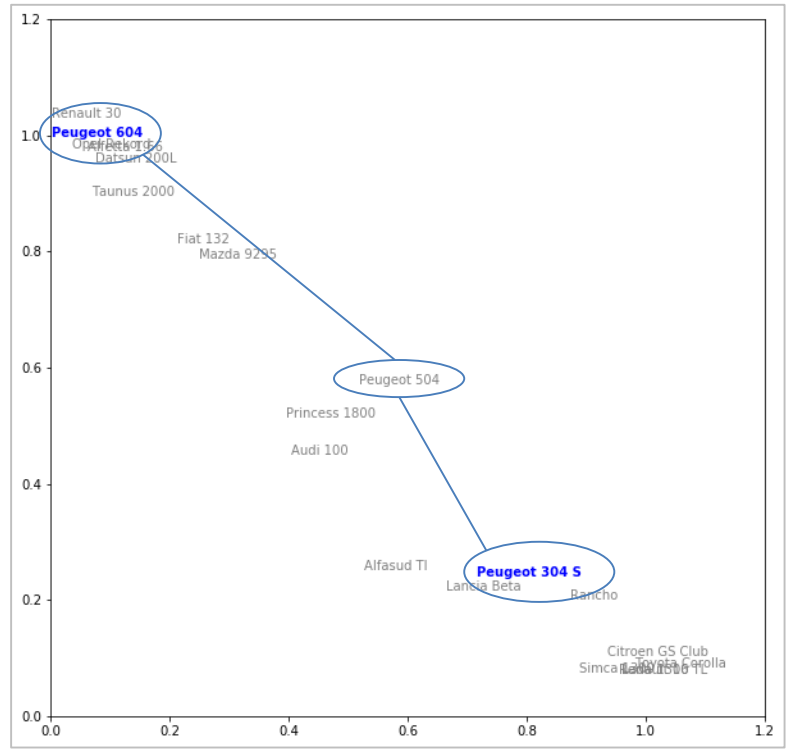
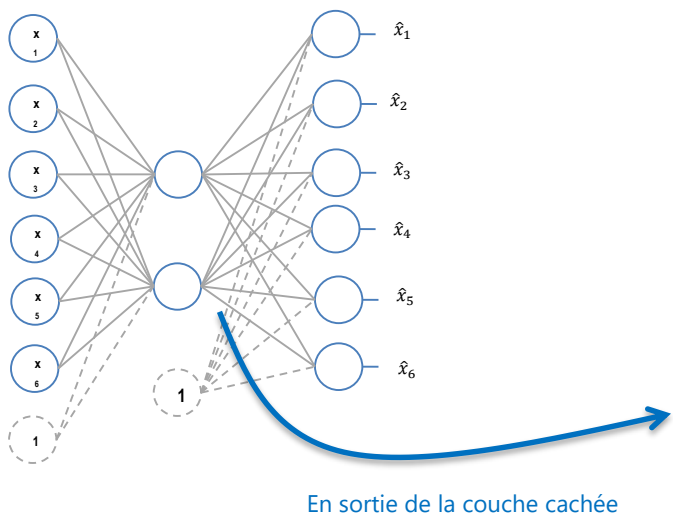
Comment les positionner par rapport aux véhicules actifs ?

Modele	CYL	PUISS	LONG	LARG	POIDS	V.MAX
Peugeot 604	2664	136	472	177	1410	180
Peugeot 304 S	1288	74	414	157	915	160



Appliquer le réseau sur ces individus

Transformer (éventuellement) les variables avant de leur appliquer le réseau, avec les paramètres calculés sur les individus actifs (ex. moyenne et écart-type si standardisation)



Le positionnement des véhicules se comprend aisément (si on connaît un peu les véhicules des années 70)



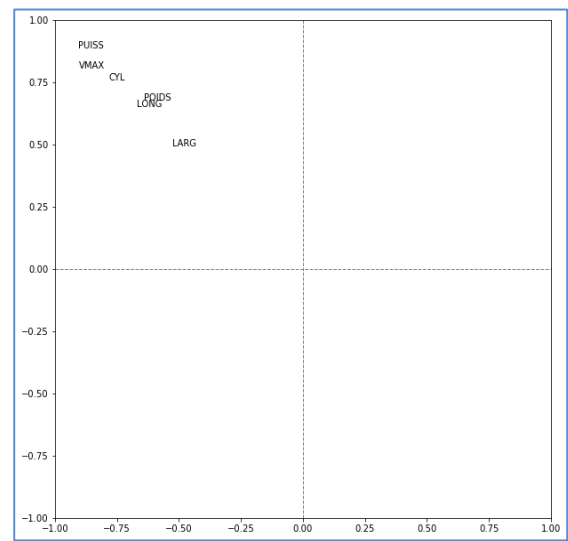
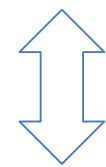
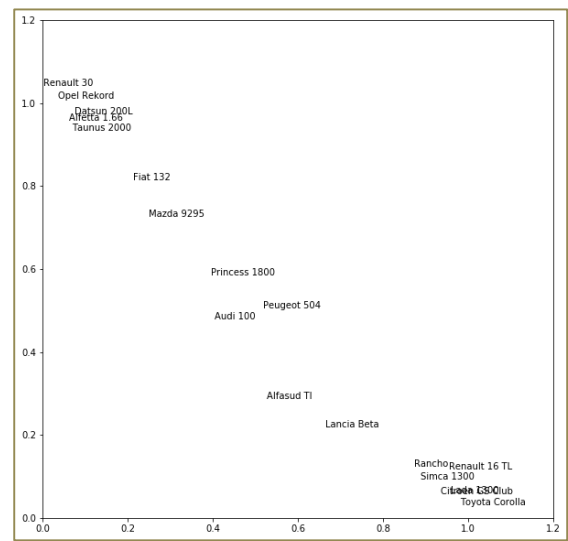
# Identifier le rôle des variables

**Valide que si les « pattern » ne sont pas (trop) non-linéaires**

Calculer les corrélations des variables  $(X_{ij})$  ( $i = 1, \dots, n ; j = 1, \dots, p$ ) avec les coordonnées « factorielles »  $(F_{ik})$  ( $k = 1, 2$ )

	Corr_1	Corr_2
CYL	-0.781299	0.828374
PUISS	-0.907450	0.906131
LONG	-0.669335	0.698577
LARG	-0.527321	0.499947
POIDS	-0.641733	0.687519
VMAX	-0.904271	0.851150

On peut imaginer une sorte de « cercle des corrélations ».



# Traitement des variables supplémentaires

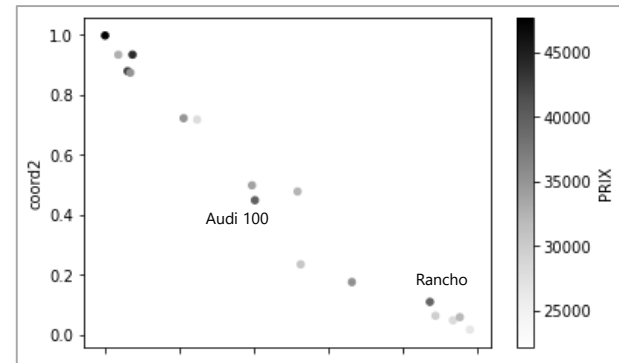
Interpréter les résultats à l'aune des variables illustratives

Modele	FINITION	PRIX	RPOIDPUIS
Alfasud TI	2_B	30570	11.01
Audi 100	3_TB	39990	13.06
Simca 1300	1_M	29600	15.44
Citroen GS Club	1_M	28250	15.76
Fiat 132	2_B	34900	11.28
Lancia Beta	3_TB	35480	13.17
Peugeot 504	2_B	32300	14.68
Renault 16 TL	2_B	32000	18.36
Renault 30	3_TB	47700	10.31
Toyota Corolla	1_M	26540	14.82
Alfetta-1.66	3_TB	42395	9.72
Princess-1800	2_B	33990	14.15
Datsun-200L	3_TB	43980	11.91
Taunus-2000	2_B	35010	11.02
Rancho	3_TB	39450	14.11
Mazda-9295	1_M	27900	13.19
Opel-Rekord	2_B	32700	11.20
Lada-1300	1_M	22100	14.04

## Variables quantitatives :

calculer les corrélations  
est une solution simple

	coord1	coord2	PRIX	RPOIDPUIS
coord1	1.000000	-0.983089	-0.623951	0.825975
coord2	-0.983089	1.000000	0.620979	-0.767637
PRIX	-0.623951	0.620979	1.000000	-0.524179
RPOIDPUIS	0.825975	-0.767637	-0.524179	1.000000

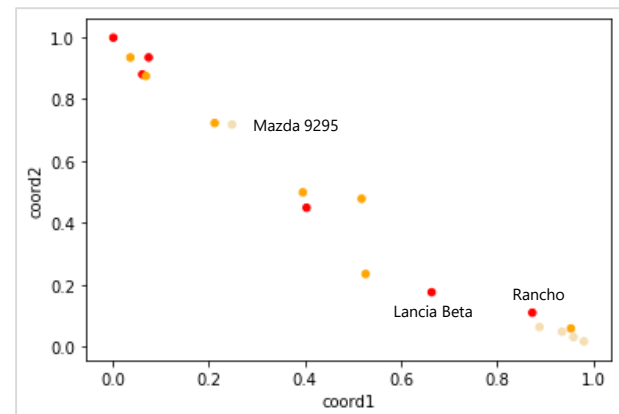


Ou encore un graphique  
(teinte des points en fonction du PRIX)

## Variables qualitatives :

calculer les moyennes conditionnelles

FINITION	coord1	coord2
1_M	0.802112	0.176284
2_B	0.387250	0.543158
3_TB	0.346295	0.591172



Ou encore un graphique  
(couleur des points en fonction de FINITION)



Spécificité du traitement d'images avec les auto-encodeurs

# TRAITEMENT D'IMAGES

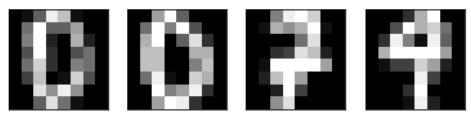


# Format des données en entrée – Opérateurs spécifiques

Les auto-encodeurs, comme toutes méthodes de deep learning, se prêtent bien au traitement d'images.

En niveau de gris, l'image peut être linéarisé en un vecteur de valeurs (compris entre 0 et 255 habituellement). On est dans le cadre habituel.

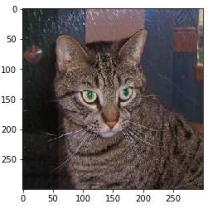
chiffre	pix1	pix2	pix3	pix4	...	pix61	pix62	pix63	pix64
C0	0	1	6	15	...	7	1	0	0
C0	0	0	10	16	...	15	3	0	0
C7	0	0	8	15	...	0	0	0	0
C4	0	0	0	3	...	15	2	0	0



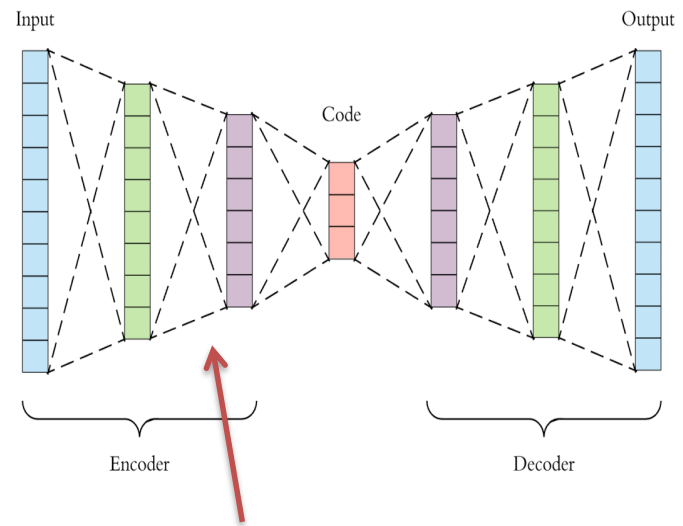
Ex. [Optical handwritten digits](#), images (8 x 8)

En couleur, elle se présente sous une forme matricielle avec 3 canaux. Les librairies spécialisées (ex. Keras) savent les appréhender.

```
#Librairie pour IO Image
import imageio
#charger l'image
chat = imageio.imread("cat_1.jpg")
print(chat.shape) #(300, 300, 3)
#Librairie pour graphiques
import matplotlib.pyplot as plt
#affichage de l'objet
plt.imshow(chat)
```



Ex. [Cats and Dogs](#), images (300 x 300 x 3)

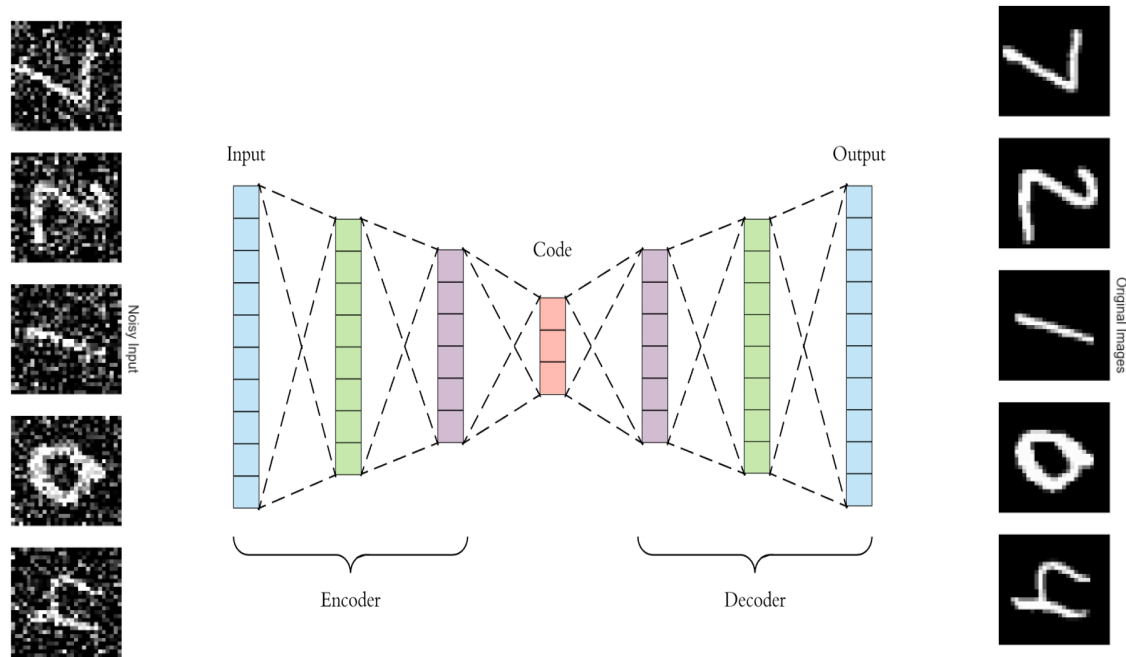


On peut appliquer les opérateurs que l'on retrouve dans les [réseaux de neurones convolutifs](#) : [convolution](#), [pooling](#), [subsampling](#)...



# Auto-encodeur débruiteur

On peut aussi voir un auto-encodeur sous l'angle d'un apprentissage multi-supervisé (plusieurs variables cibles). On peut améliorer la robustesse du « modèle » en perturbant (en ajoutant du bruit) les entrées tout en conservant une sortie de référence sans bruit (pour calculer la fonction de perte, ex. MSE).



Images de référence pour l'entrée

Images de référence pour la sortie, à confronter avec l'output du réseau et calculer le MSE.



Les propriétés de « nettoyage » de l'auto-encodeur joue à plein pour extraire l'information essentielle.





# CONCLUSION



- L'auto-encodeur est une technique de deep learning pour la réduction de la dimensionalité et la représentation des données
- Il se présente comme un perceptron multicouches non-supervisé (ou multi-supervisé, c'est selon le point de vue)
- Dans certaines configurations, il produit des résultats similaires à ceux de l'ACP (analyse en composantes principales)
- Mais, à la différence de l'ACP, il peut proposer différents niveaux d'abstraction et est capable de restituer des « pattern » non-linéaires
- Amélioration de la robustesse par utilisation de données sciemment « bruitées »
- Il est adapté au traitement d'images en sachant appréhender les données matricielles en entrée (via des bibliothèques spécialisées).



# RÉFÉRENCES



- Wikipédia, « [Autoencoder](#) », consulté le 27.11.2019.
- Dertat A., « [Applied Deep Learning – Part 3 : Autoencoders](#) », Towards Data Science, Oct. 2017.
- Cohen O., « [PCA vs Autoencoders](#) », Towards Data Science, Avr. 2018.
- Cholet F., « [Building Autoencoders in Keras](#) », The Keras Blog, Mai 2016.
- Rakotomalala R., « [Perceptrons simples et multicouches](#) », Nov. 2018.

