

# Deep learning : Auto-encodeurs

Nabil Redha BELKHOUS

Numéro d'étudiant : 16705491

Email : nbelkhous@gmail.com

Université Paris 8 Vincennes-Saint-Denis

UFR : MIASHS

Master 2 Big Data et fouille de données

Année Universitaire : 2016 - 2017

# Sommaire

Introduction

Auto-encodeurs

Famille d'auto-encodeurs

Type d'auto-encodeurs

Deep auto-encodeurs

Conclusion

# Introduction



# Deep learning

## Description [8]

- Le deep learning est une méthode basée sur le machine learning, plus précisément sur un réseau de neurones multi-couches.
- Il permet de découvrir des structures complexes dans de grands ensembles de données.
- Il utilise les méthodes de Rétro-propagation pour indiquer comment changer les paramètres afin calculer la représentation d'une couche à partir de sa couche précédente.

# Auto-encodeurs

## Lors de la présentation

Je parlerai dans un premier temps des:

- auto-encodeurs basiques
- leurs caractéristiques
- certains types d'auto-encodeurs

Puis, dans un deuxième temps je parlerai de l'utilisation des auto-encodeurs dans le cadre du deep learning.

# Auto-encodeurs

## 2. Auto-Encodeurs

# Généralités

## Description

- Réseau de neurones
- Apprentissage non supervisé
- feed-forward [2]
- Trois couches

## Principe

Les auto-encodeurs sont entraînés à reproduire l'input en sortie du réseau.

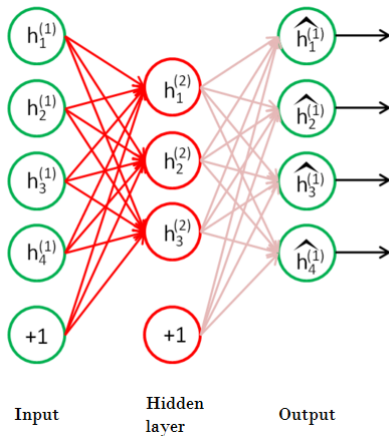
# Architecture

## Description

- **Input** : les données en entrée du réseau
- **Hidden layer** : Input sous format compressé
- **Output** : Reconstitution des données



## Architecture

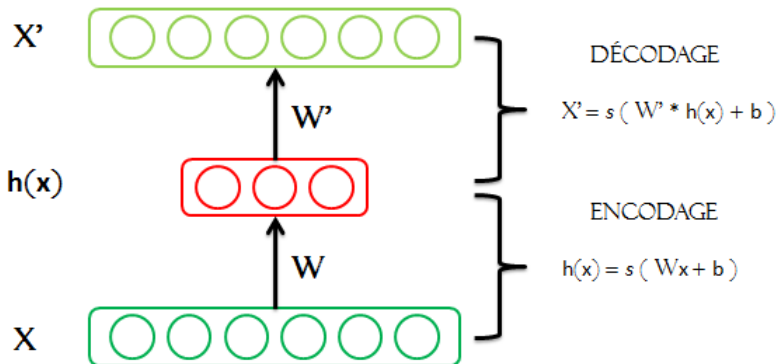


# Encodage - Decodage

## Description

Les auto-encodeurs compressent les données entrées dans le réseau, extraient les caractéristiques les plus intéressantes puis décompressent les données.

## Encodage - Decodage



[6]

## Phase d'entraînement

### Méthode utilisée

- Déterminer une **Loss Function[6]** (qui va comparer l'input et l'output de notre réseau pour mesurer la qualité de la reconstruction)
- Entraîner le réseau à minimiser le résultat de la **Loss fucntion** en utilisant la méthode de **rétro-propagation du gradient descendant**.

## Loss Function

### Loss Function

La **Loss function** est spécifiée selon le type de données[6]. Par exemple:

- pour les données binaires la *lossFunction* est définie par:

$$lossFunction(Y) = - \sum_k (x_k \times \log(x'_k)) + (1 - x_k) \times \log(1 - x'_k)$$

- pour les données réelles la *lossFunction* est définie par:

$$lossFunction(Y) = \frac{1}{2} \sum_k (x'_k - x_k)^2$$

## Famille d'auto-encodeurs

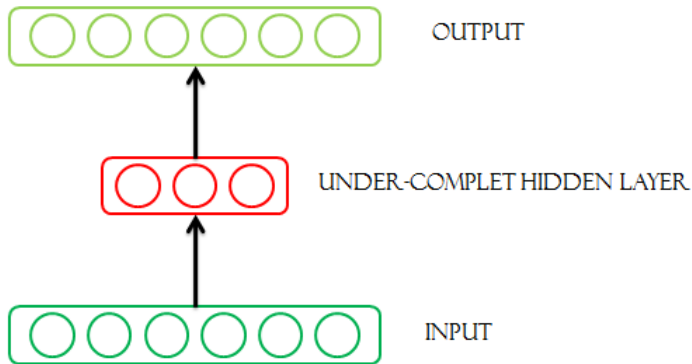
### 3. Famille d'auto-encodeurs

# Famille d'auto-encodeurs

## Deux topologies

- 1 Under-complet hidden layer
- 2 Over-complet hidden layer

## Under-complet hidden layer



[3][4]

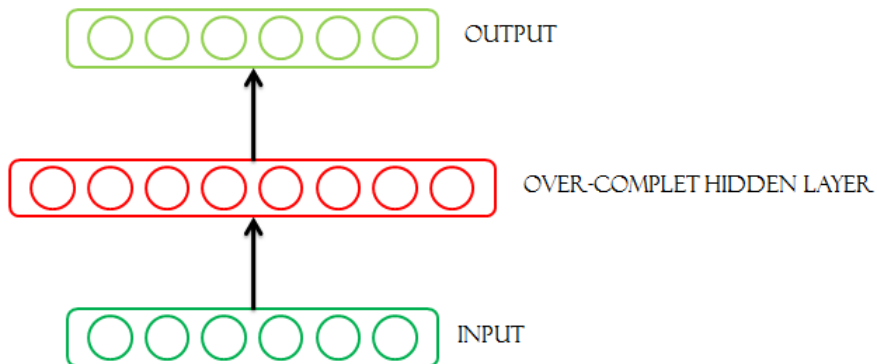


## Under-complet hidden layer

### Caractéristiques

- Couche cachée plus petite que l'input
- Couche cachée compresse l'input
- Réseau entraîné uniquement sur les données de l'ensemble d'entraînement
- Unités de la couche cachée extraient de bonnes caractéristiques à partir de l'ensemble d'entraînement.
- Topologie donne de mauvais résultats avec des données hors de l'ensemble d'entraînement.

## Over-complet hidden layer



## Over-complet hidden layer

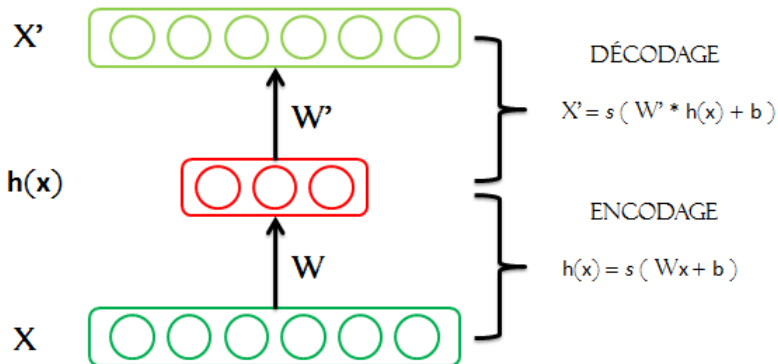
### Caractéristiques

- Couche cachée plus grande que l'input
- Couche cachée ne compresse pas l'input
- Réseau entraîné uniquement sur les données de l'ensemble d'entraînement
- Unités de la couche cachée sont soit une copie de l'input ou des données ajoutées aléatoirement
- Méthode qui ne garantit pas que les unités des couches cachées vont extraire des structures significatives

## Type d'auto-encodeurs

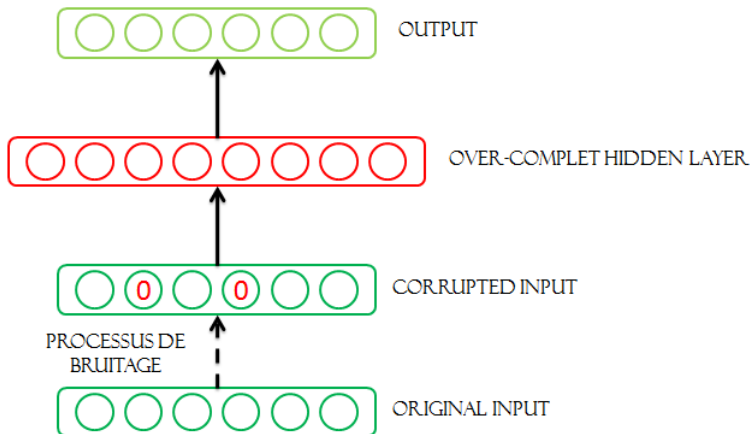
### 4. Type d'auto-encodeurs

## Auto-encodeur basique



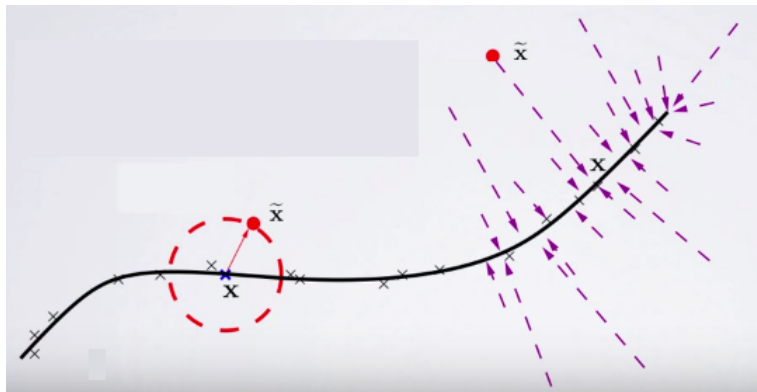
[6]

## Denoising auto-encodeur



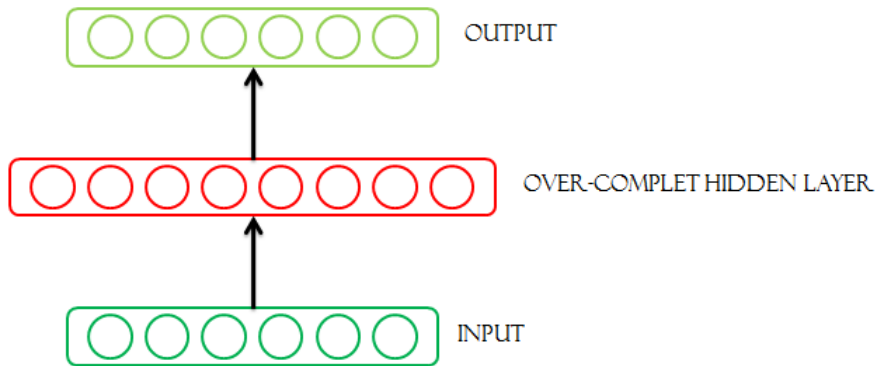
[6]

## Denoising auto-encodeur



[6]

## Contactive auto-encodeur



[7]



## Contactive auto-encodeur

### Loss Function[7]

Une nouvelle Loss function la *NewLossFunc* est définie:

$$NewLossFunc = \text{lossFunction}(Y) + \lambda \|\nabla_{x^{(t)}} h(x^{(t)})\|_F^2$$

- la loss function de l'auto-encodeur basique:

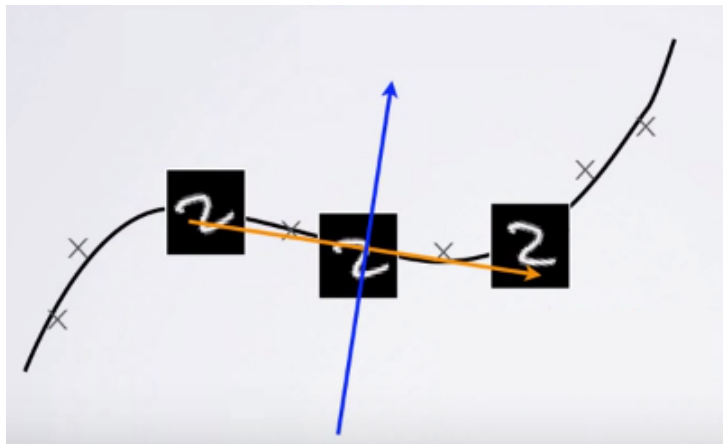
$$\text{lossFunction}(Y) = - \sum_k (x_k \times \log(x'_k)) + (1 - x_k) \times \log(1 - x'_k)$$

- et la jacobienne de l'encodage ( $H(x)$ ) :

$$\|\nabla_{x^{(t)}} h(x^{(t)})\|_F^2 = \sum_j \sum_k \left( \frac{\partial h(x^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

C'est l'addition de la Loss Function et la jacobienne qui va nous permettre de garder uniquement les caractéristiques qui reflètent les variations observées dans les données.

## Contactive auto-encodeur



[6]

## Deep auto-encodeurs

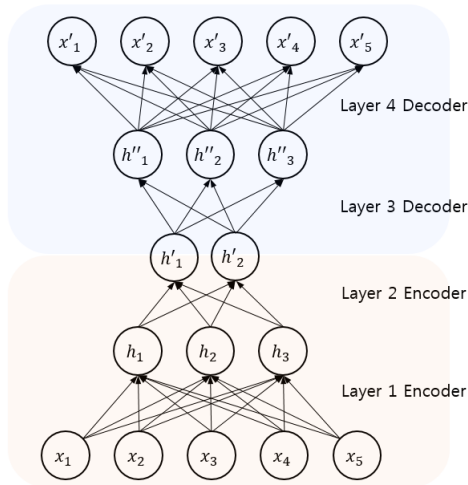
# 5. Deep auto-encodeurs

# Principe et Architecture

## Principe

Le deep auto-encoder network ou réseau d'auto-encodeur profond est un apprentissage non supervisé qui n'est qu'un auto-encodeur multi-couches. [5]

# Principe et Architecture



[5]

## Problèmes et solutions

### ① Volumétrie des données

#### Solutions proposées

- Principal Component Analysis PCA
- Auto-encodeurs

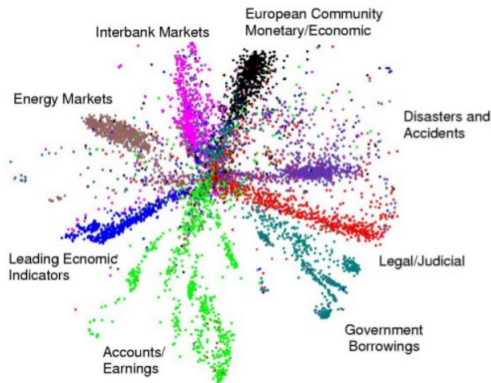
## Problèmes et solutions

### ① Volumétrie des données



# Problèmes et solutions

## ① Volumétrie des données



[1]



## Problèmes et solutions

### ② Entraînement du réseau

Les auto-encodeurs sont souvent entraînés en utilisant des méthodes de back-propagation.

Malgré son efficacité, il y a des problèmes avec son utilisation car elle est dépendante de la phase d'initialisation des poids.

Lors de l'entraînement du réseau avec plusieurs couches cachées, les erreurs sont retropropagées aux premières couches et elles deviennent petites et l'entraînement est inefficace.[1]

## Problèmes et solutions

### ② Entraînement du réseau

#### Problème

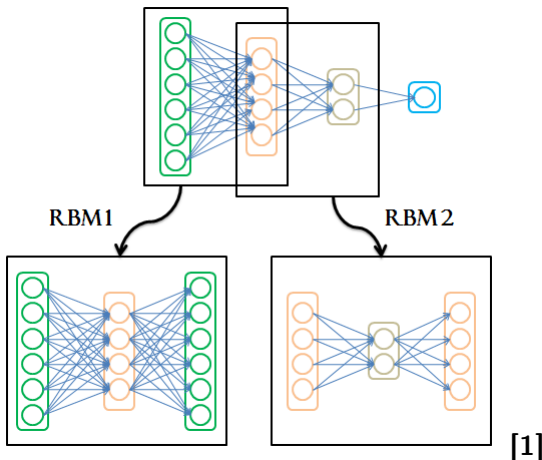
- Back-propagation peu efficace

#### Solutions proposées

- Phase de pré-entraînement

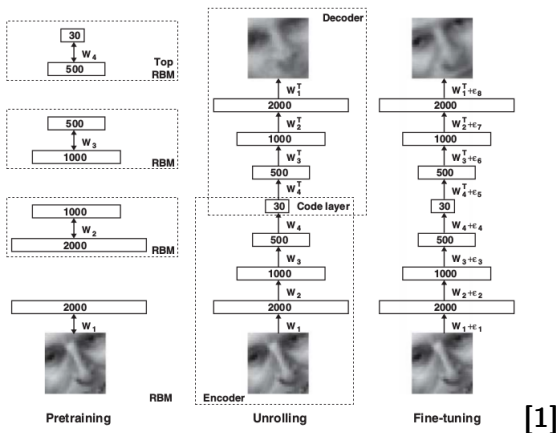
## Problèmes et solutions

### ② Entraînement du réseau



# Problèmes et solutions

## 2 Entraînement du réseau



## Conclusion

# Conclusion

## Conclusion

A travers mes recherches j'ai pu constaté que les auto-encodeurs :

- Apprentissage automatique
- Feed-forward
- Basic, denoising, contractive, sparse, transforming, ...
- Under-complet hidden layer et Over-complet hidden layer
- Rétro-propagation
- Pré-entraînement
- Apprentissage sur des données multi-dimensionnelles
- Meilleurs résultats que le Principal Component Analysis pour le clustering

## Livres et articles: I



[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville.  
*Deep learning. Book in preparation for MIT.*  
Press, 2016.



[5] Dong Yu and Li Deng  
*Deep Learning : Methods and Applications.*  
Foundations and Trends in Signal Processing : Vol. 7, 2014



[1] R. Salakhutdinov G. E. Hinton.  
*Reducing the dimensionality of data with neural networks*  
pages 504–507, 2006.

## Livres et articles: II



[2] S. D. Wang G. E. Hinton, A. Krizhevsky.

*Transforming Auto-encoders.*

ICANN-11 : International Conference on Artificial Neural Networks, Helsinki., 2011.



[6] I. Lajoie Y. Bengio P.-A. Manzagol P. Vincent, H.

Larochelle.

*Stacked Denoising Autoencoders : Learning Useful Representations in a Deep Network with a Local Denoising Criterion.*

Journal of Machine Learning Research, vol. 11, pages 3371–3408, 2010.



## Livres et articles: III



[7] Xavier Muller Xavier Glorot Yoshua Bengio Salah Rifai, Pascal Vincent.

*Contractive auto-encoders : Explicit invariance during feature extraction.* , 2011



[8] Geoffrey Hinton Yann LeCun, Yoshua Bengio.

*Deep learning.*

REVIEW INSIGHT : Annals of Mathematical Logic, pages 436–442, 2015.



[4] Hugo Larochelle. Neural networks.

*Autoencoder - undercomplete vs. overcomplete hidden layer*  
YouTube, 2013

# Questions

QUESTIONS ?