Visualisation de données massives par réduction de dimension non linéaire

Maxime Laborde, Clément Guerin, Mariette Dupuy

Université de Bordeaux

24 Mars 2020

Sommaire

1. Introduction
2. Principes de l'auto encoder
3. Explication ACP
4. Mise en situation sur l'auto-encoder et l'ACP

Sommaire

1. Introduction
2. Principes de l'auto encoder
3. Explication ACP
4. Mise en situation sur l'auto-encoder et l'ACP

Introduction

- Présence de machine learning et du Deep learning de plus en plus important
- Comportement des algorithmes avec des paramètres spécifiques
- Auto encodeurs $\rightarrow$ réseaux de neurones artificiels
- Efficacité d'une descente en dimension
- Comparaison des résultats ACP et Auto Encoder

Sommaire

1 Introduction

2 Principes de l'auto encoder

3 Explication ACP

4 Mise en situation sur l'auto-encoder et l'ACP

Principes de l'auto encoder

Dans toute la suite nous considérerons nos données sous la forme (X_i, Y_i) avec i$\in 1, ..., n$

- $\mathrm{X}_1, ..., X_n \in \mathbb{R}^p$ avec p la dimension de nos données (p=4 dans le cas des iris)

- $\mathrm{Y}_i \in 1, 2, ..., k$ un ensemble de classe appelé l'étiquette de nos données.

Principes de l'auto encoder

- Extraire des données deux ensemble : un ensemble d'apprentissage et un ensemble test
- Réseau de Neurones $\rightarrow$ plusieurs couches de neurones $\rightarrow$ règle de passage $\rightarrow$ fonctions d'activation
- "fully connected" $\rightarrow$ connecte tous les neurones d'une couche à tous ceux de la suivante
- Nombre d'epochs $\rightarrow$ représente le nombre de fois que l'on souhaite que l'algorithme parcourt l'ensemble des données.
- Nombre de Batch $\rightarrow$ représente le nombre de données ou d'échantillons de données qu'il convient de parcourir avant de mettre à jour les paramètres du modèle

Ensemble d'apprentissage

L'algorithme va appliquer une certaine fonction à nos données et leur rajouter un biais :

$$X_i \rightarrow \sigma(W_1 X_i + b_1)$$

Avec W1 taille d x p et $b_1 \in \mathbb{R}^d$

Passage de dimension p en dimension d (Iris d=2)

Nombre de répétitions de ce processus = Nombre couche de neurones

Dans la suite de la description du principe général nous supposerons qu'il n'y a qu'une couche d'encodage.

Decodage

Pour cela de manière analogue à la partie précédente nous allons transformer les données obtenues à l'issue de l'encodage en appliquant de nouveau une fonction puis un biais.

$$\sigma(W_1 X_i + b_1) \rightarrow \sigma(W_2 \sigma(W_1 X_i + b_1) + b_2)$$

Avec W_2 de dimension p x d (p=4 dimension initiale), et $b_2 \in \mathbb{R}^p$ représente le biais.

Decodage

On peut réécrire cette équation sous la forme :

$$WX_i + b$$

Avec $W = W_2 W_1$ et $b = W_2 b_1 + b_2$

A l'issue de cette étape nous avons donc les données reconstruites.

Calcul d'erreur

Mean Squared Error $\rightarrow$ minimiser la fonction suivante :

$$\frac{1}{n}\sum_{i=1}^{n}\left\|W_2(W_1X_i+b_1)+b_2-X_i\right\|^2$$
$$\Longleftrightarrow$$
$$\frac{1}{n}\sum_{i=1}^{n}\left\|WX_i+b-X_i\right\|^2$$

Cette fonction est à minimiser sur ses arguments W_1, W_2, b_1 et b_2.

Sommaire

1 Introduction

2 Principes de l'auto encoder

3 Explication ACP

4 Mise en situation sur l'auto-encoder et l'ACP

Une première manière intuitive de voir l'ACP

Si d=2 on a :

$$w^1 \in \mathbb{R}^p \quad w^2 \in \mathbb{R}^p$$

avec $\langle w^1, w^2 \rangle = 0$ car ils doivent être orthogonaux

La formule à maximiser est la suivante :

$$\max_{w^1} \frac{1}{n} \sum_{i=1}^{n} \langle Xi, w^1 \rangle^2$$

Or on a que :

$$\langle Xi, w^1 \rangle^2 \Longleftrightarrow \mathrm{w}^{1t} X i X i^t w^1$$

Une première manière intuitive de voir l'ACP

En conséquence nous avons que

$$\max_{w^1} \frac{1}{n} \sum_{i=1}^{n} \langle Xi, w^1 \rangle^2 \iff \max_{w^1} \frac{1}{n} \sum_{i=1}^{n} w^{1t} Xi Xi^t w^1 \iff \max_{w^1} w^{1t} \Sigma w^1$$

avec Σ la matrice de covariance

Une autre manière de voir l'ACP qui permet de faire apparaître le lien avec les AE

L'objectif est de minimiser l'expression suivante :

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\langle Xi, w^1\rangle w^1 + \langle Xi, w^2\rangle w^2 - Xi\right\|^2$$

$$\Longleftrightarrow$$

...

$$\Longleftrightarrow$$

$$\max_{w^1} w^{1t}\Sigma w^1 + \max_{w^2} w^{1t}\Sigma w^2$$

Lien entre AutoEncoder et ACP

Une de nos problématiques est de démontrer l'équivalence suivante :

$$\text{Autoencoder} \iff \text{ACP}$$

Pour observer un lien entre ces méthodes il est important de simplifier la deuxième formule de l'ACP.

$$\left\| \langle Xi, w^1 \rangle w^1 + \langle Xi, w^2 \rangle w^2 - Xi \right\|^2$$
$$\iff$$
$$\left\| w^1 w^{1t} Xi + w^2 w^{2t} Xi - Xi \right\|^2$$
$$\iff$$
$$\left\| (w^1 w^{1t} + w^2 w^{2t}) Xi - Xi \right\|^2$$

On pose $W = w^1 w^{1t} + w^2 w^{2t}$. Avec W une matrice de rang 2.

Lien entre AutoEncoder et ACP

On en conclut donc qu'il faut minimiser dans cette version la formule suivante :

$$\frac{1}{n}\sum_{i=1}^{n}\|WXi - Xi\|^2$$

L'AutoEncoder lui a pour objectif de minimiser la formule suivante :

$$\frac{1}{n}\sum_{i=1}^{n}\|WXi + b - Xi\|^2.$$

On peut donc constater que les deux formules sont similaires à un biais près.

Lien entre AutoEncoder et ACP

Optimisation :

L' ACP optimise sur des espaces très contraints $\neq$ l'AutoEncoder optimise sans contrainte

De plus l'Auto encoder, ne rencontre aucun problème d'orthogonalité.

AutoEncoder	ACP
rang=1 ou rang=2	rang=2

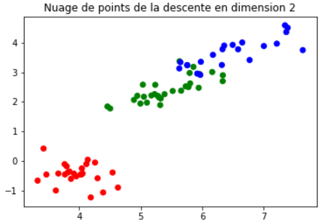
Rang(W) = 2 : permet d'observer une équivalence entre l'ACP et l'AE

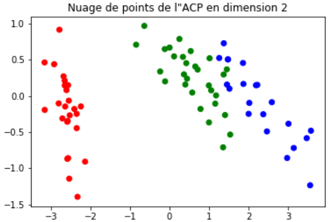
Sommaire

1. Introduction
2. Principes de l'auto encoder
3. Explication ACP
4. Mise en situation sur l'auto-encoder et l'ACP

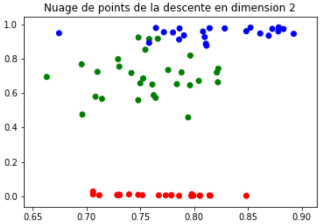
Jeu de donnée n°1 : Iris

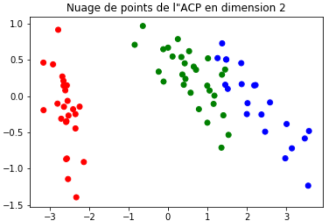
Fonctions d'activation de type "linear" en entrée et en sortie





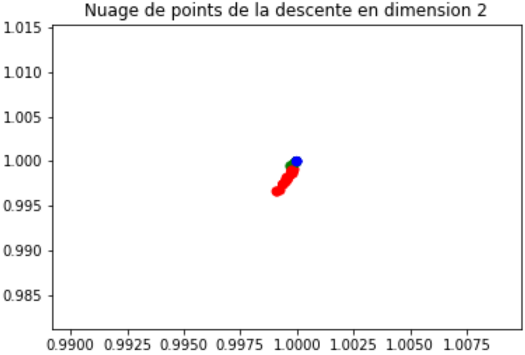
Fonction d'activation d'entrée par la fonction "sigmoid"





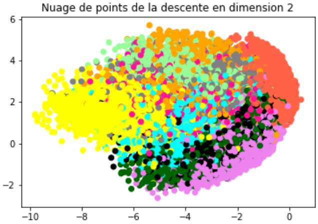
Jeu de donnée n°1 : Iris

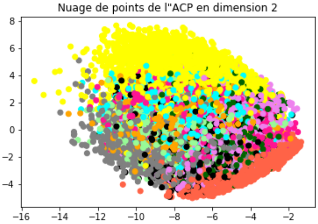
Fonction d'activation "sigmoid" en entrée et en sortie

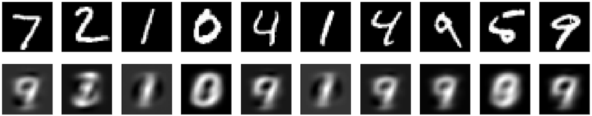


Jeu de donnée n°2 : MNIST

Fonction d'activation "linear" en entrée et en sortie

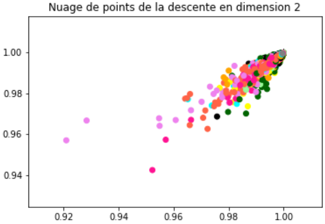


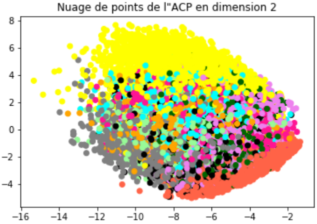


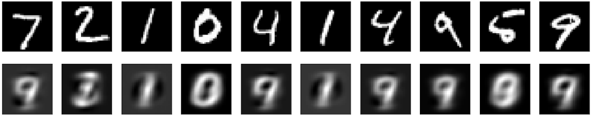


Jeu de donnée n°2 : MNIST

Fonctions d'activation "sigmoid" en entrée et en sortie

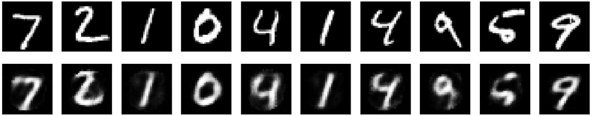






Jeu de donnée n°2 : MNIST

Descente en dimension 32 avec une fonction "ReLU" en entrée et "sigmoid" en sortie :



Nous obtenons une perte de 0.02 , ce qui est un très bon résultat comme on peut le voir sur le graphique ci-dessus.