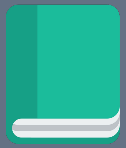
Majeure Machine Learning

Concepts

Contenu



- Principe d’apprentissage (Cost function)

- Régression linéaire + polynomiale

- Descente de gradient

- Underfitting / Overfitting

- Régularisation

- Espaces de grandes dimensions

Ce que vous devrez savoir faire

- Comprendre le rôle de la Cost function

- Comprendre le principe de minimisation de coût grâce à la descente de gradient

- Définir et identifier l’Overfitting et l’ Underfitting

- Comprendre le rôle de la Régularisation

- Comprendre les limites liées aux dimensions

Principe d’apprentissage

Qu’est-ce que l'entraînement ?

Jeu de données :

Superficie (m²)	Prix (en milliers d’€)
2000	200 000
2500	250 000
1600	160 000
1200	120 000
2300	230 000
1800	180 000

≈80%

Jeu d’entrainement :

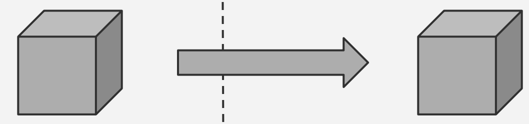
Superficie (m²)	Prix (en milliers d’€)
2000	200 000
2500	250 000
1600	160 000
1200	120 000

≈20%

Jeu de test :

Superficie (m²)	Prix (en milliers d’€)
2300	230 000
1800	180 000

233 000

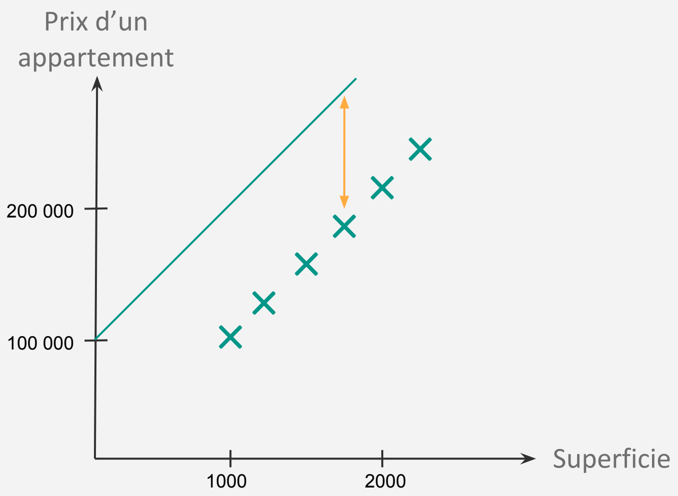


Modèle

Modèle entraîné

Cost function

Superficie (m²)	Prix (en milliers d'€)	Prix prédit
2000	200 000	300 000
2500	250 000	350 000
1600	160 000	260 000
1200	120 000	220 000
2300	230 000	330 000
1800	180 000	280 000



$$J(\theta) = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))^2$$

=> Objectif : Minimiser la Cost function

Qu’est-ce que l'entraînement ?

Jeu de données :

X	Y
x_1	y_1
x_2	y_2
…	…
x_n	y_n

Features : X
Label : Y

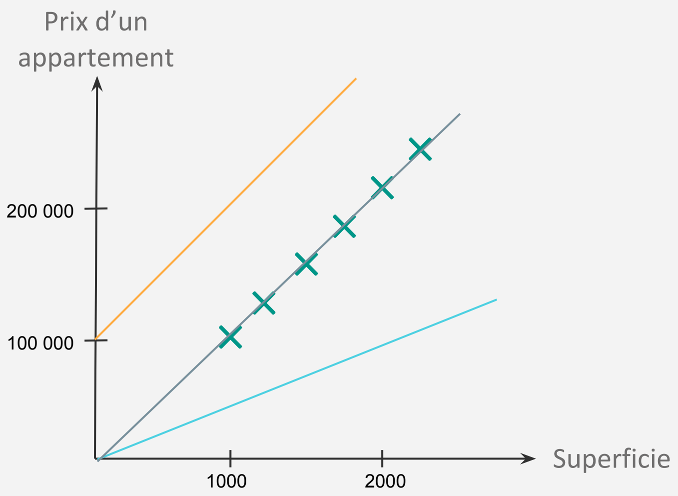
Valeur réelle : y(x)
Valeur prédite : $\hat{y}(x)$
Modèle : $\hat{y}$

Objectif :

$$\forall x_i \, , \hat{y}(x_i) \approx y(x_i)$$

Exemple : La Régression linéaire

Superficie (m²)	Prix (en milliers d' €)
2000	200 000
2500	250 000
1600	160 000
1200	120 000
2300	230 000
1800	180 000



$$(\theta_1, \theta_0) => \hat{y}(x) = \theta_1 x + \theta_0$$

(10, 0)

(100, 0)

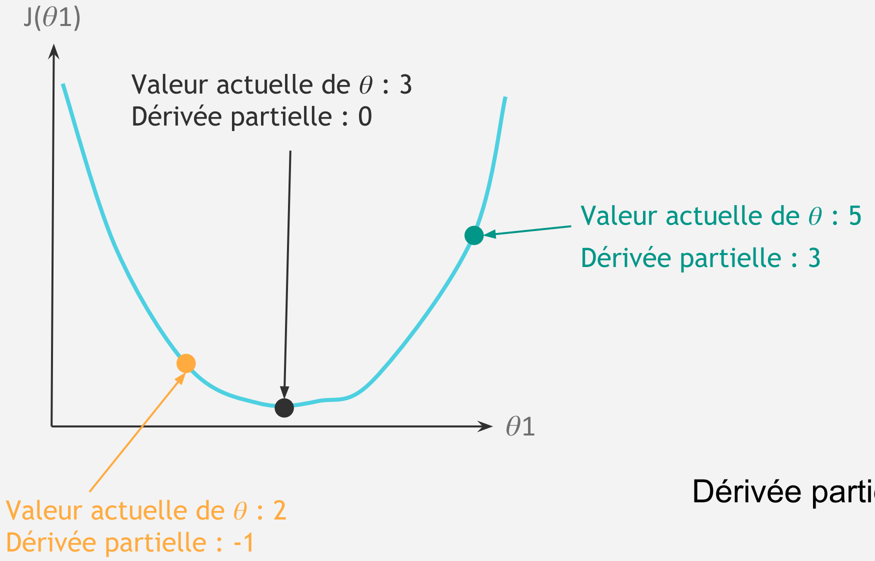
(100, 100)

=> Comment trouver θ_0 et θ_1 ?

Minimisation de la Cost function

$$J(\theta) = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))^2$$

=> Objectif : Mettre à jour itérativement θ grâce à sa dérivée partielle pour atteindre un minimum de J(θ)

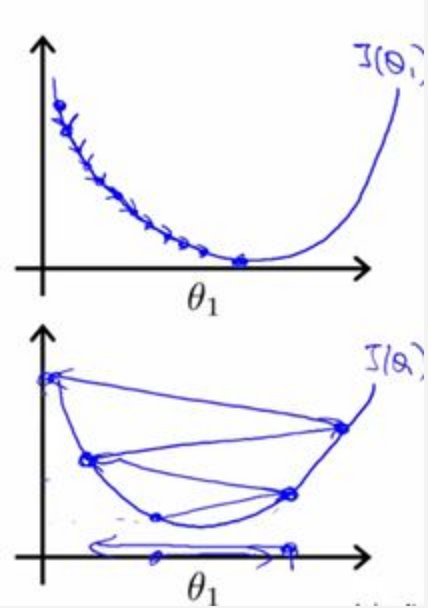


Dérivée partielle : $$\frac{\partial J(\theta)}{\partial \theta_1} = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))x_n$$

La descente de gradient

$$J(\theta) = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))^2$$

Dérivée partielle : $\dfrac{\partial J(\theta)}{\partial \theta_1} = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))x_n$



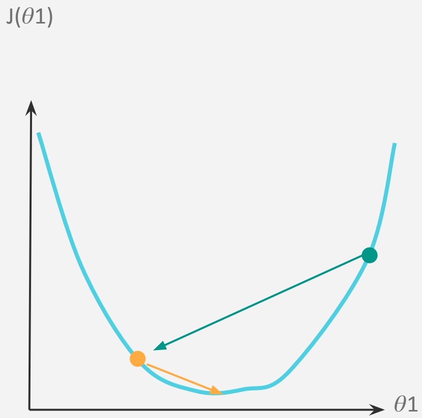
Pour i allant de 1 à nombre_choisi :

$$\theta_1 = \theta_1 - \alpha \frac{\partial J(\theta)}{\partial \theta_1}$$

Avec α > 0, le pas d'avancement

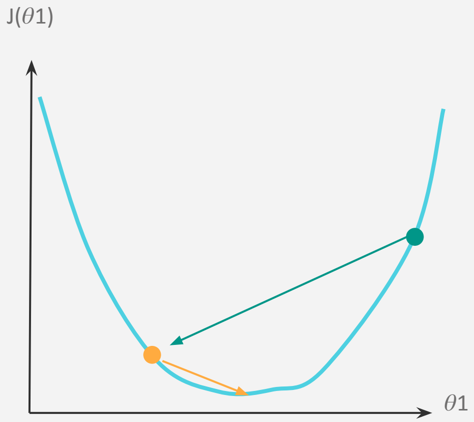
$\theta 1$ = 5
Dérivé partielle = 3
=> $\theta 1$ = 5 - 1*3 = 2

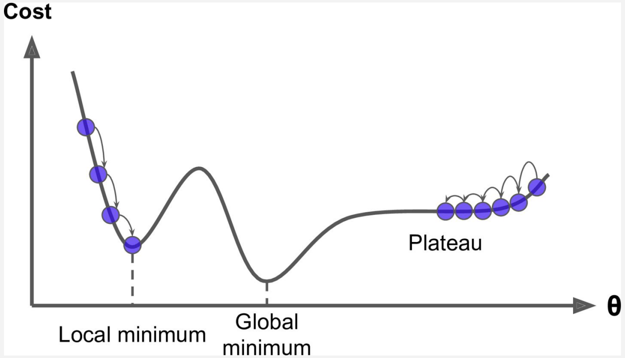
$\theta 1$ = 2
Dérivé partielle = -1
=> $\theta 1$ = 2 - 1*(-1) = 2 + 1 = 3

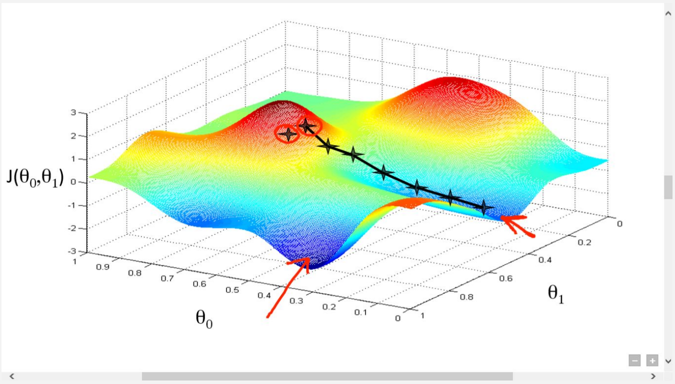


Minimisation de la Cost function

$$J(\theta) = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))^2$$

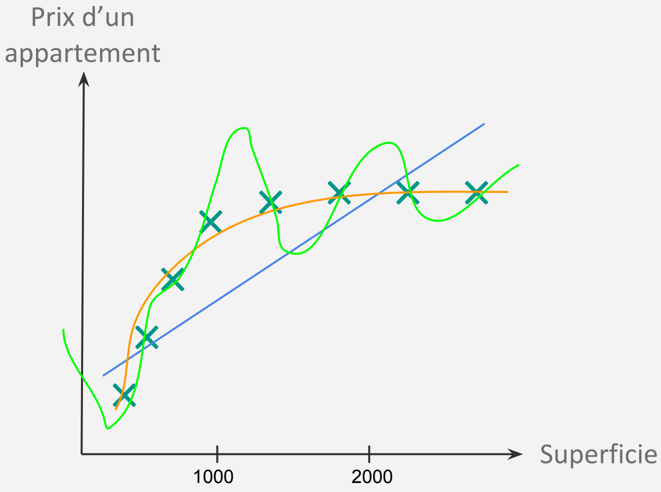






Capacité d'apprentissage

Capacité d'un modèle



$$\hat{y}(x) = \theta_1 x_1 + \theta_0$$

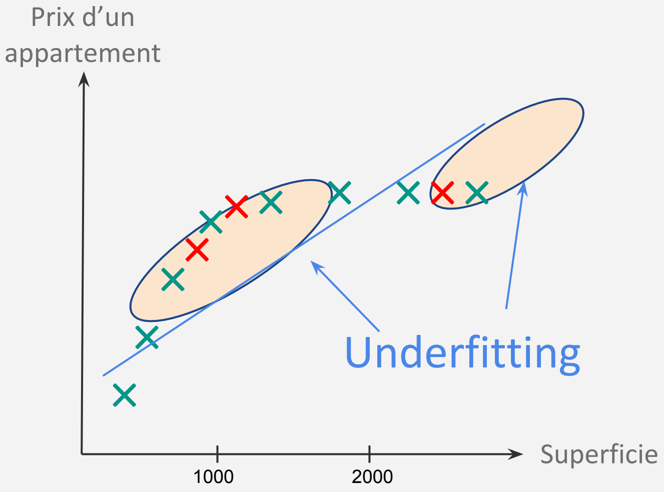
$$\hat{y}(x) = \theta_2 x^2 + \theta_1 x_1 + \theta_0$$

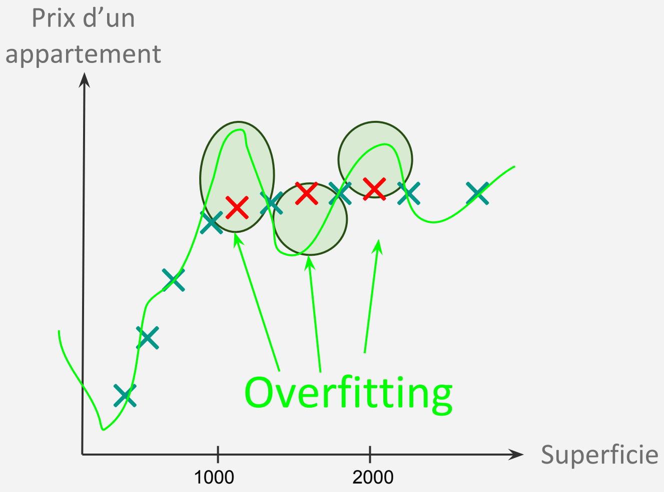
$$\hat{y}(x) = \theta_4 x^4 + \theta_2 x^3 + \theta_2 x^2 + \theta_1 x_1 + \theta_0$$

=> Capacité :

- **Faculté à apprendre "par coeur"**

Overfitting / Underfitting





$$\hat{y}(x) = \theta_1 x_1 + \theta_0$$

$$\hat{y}(x) = \theta_4 x^4 + \theta_2 x^3 + \theta_2 x^2 + \theta_1 x_1 + \theta_0$$

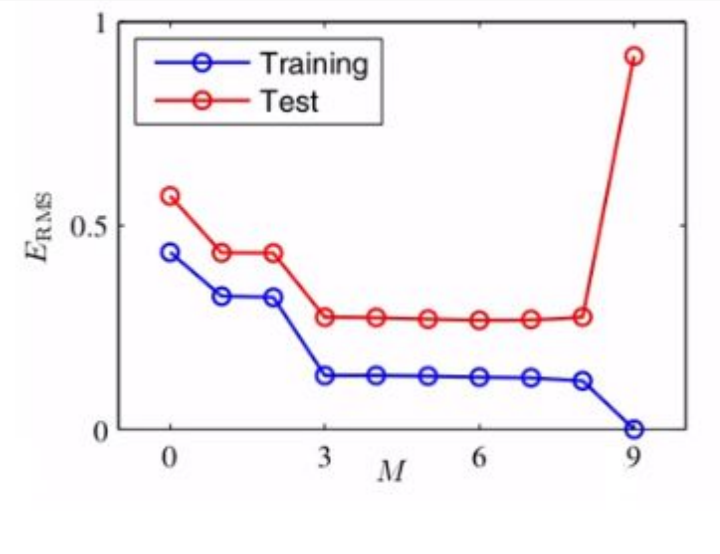
Overfitting / Underfitting

Les causes :

- Underfitting :
    - Capacité du modèle trop faible
    - Pas assez d'itérations d'apprentissage
    - Manque de qualité dans les données

- Overfitting :
    - Capacité du modèle trop forte
    - Trop peu d'exemples

Diagnostique :



Overfitting / Underfitting - Quizz

	Overfitting	Underfitting
Train : 95% | Test : 60%	✓	
Train : 80% | Test : 78%		
Train : 40% | Test : 42%		✓

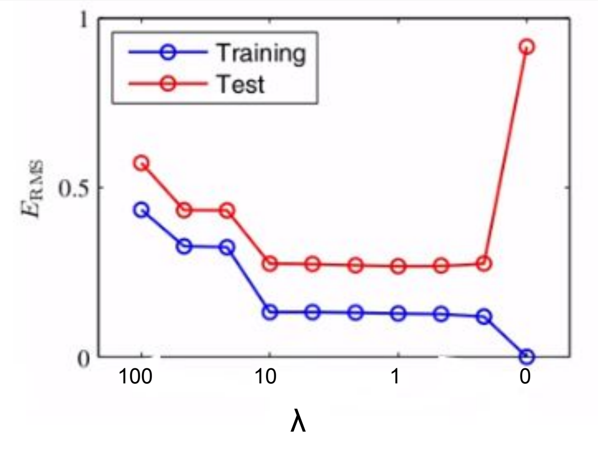
La Régularisation

=> Objectif : Réduire la capacité pour prévenir l'Overfitting

Pénalisation de la Cost function :

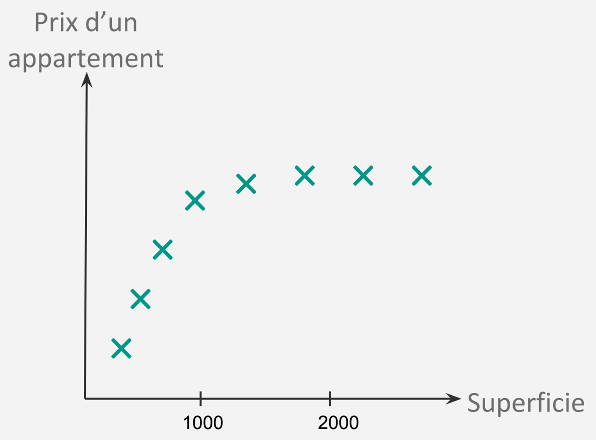
$$J(\theta) = 1/N \sum_{n=1}^{N} (\hat{y}(x_n, \theta) - y(x_n))^2 \quad + \lambda\theta^2$$

Avec λ > 0, la "force" de régularisation

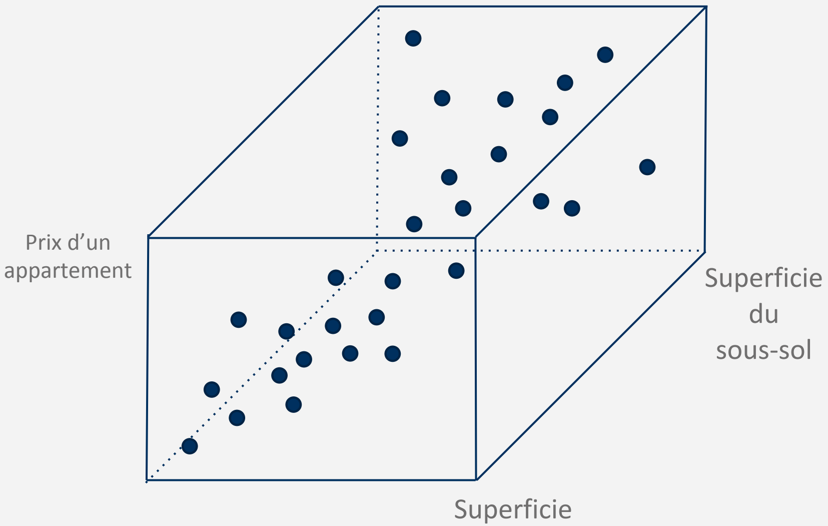


Les problèmes liés aux grandes dimensions

La Malédiction des dimensions



+1 feature



=> La <u>difficulté à généraliser</u> peut augmenter exponentiellement avec le nombre de dimensions

=> La complexité et le <u>temps de calcul</u> vont aussi augmenter

Fin du chapitre 2.1