Comment enseigne-t-on aux algorithmes?

Régression Linéaire et Logistique

Martin Vielvoye 2021

Cost Function

La Régression Linéaire

Hypothèse et Précision

Comment évaluer les performances d'un algorithme ?

- Une hypothèse est une fonction (h) qui prend un ensemble X ($x_j \forall j \in \mathbb{N}$) de variables et qui sort une *prédiction* ($\hat{y}$).

➡ Notre première hypothèse !

$$h(X) = \hat{y}$$

h	:	l'hypothèse
X	:	les variables
$\hat{y}$	:	la prédiction

Hypothèse et Précision

Comment évaluer les performances d'un algorithme ?

- La distance entre une *prédiction* ($\hat{y}$) et l'*objectif* (y) est une bonne façon d'évaluer le comportement d'un algorithme.
- On appellera l'évaluation finale le *coût* (J).
- Plus une *prédiction* est loin de l'*objectif*, plus on aimerait que le *coût* soit grand.

➡ Pour calculer la distance entre deux valeurs : <u>La soustraction</u>

$$J = \hat{y} - y$$

J	:	le coût
$\hat{y}$	:	la prédiction
y	:	l'objectif

Hypothèse et Précision

Comment évaluer les performances d'un algorithme ?

- Une relation linéaire est trop faible. On voudrait un *coût* beaucoup plus fort pour les *prédictions* fortement éloignées.

➡ Donner un rapport logarithmique à la distance : Mettre au carré

$$J = (\hat{y} - y)^2$$

J : le coût
$\hat{y}$: la prédiction
y : l'objectif

Hypothèse et Précision

Comment évaluer les performances d'un algorithme ?

- On évalue un algorithme sur l'ensemble de ses prédictions et non une seule.
- On fait donc une moyenne des prédictions pour chaque donnée x^i $(\forall i \in [0,m])$. On dit qu'il y a m valeurs d'entrainement.

➡ Pour faire une moyenne, on somme tous les coûts et on divise par le nombre de valeurs.

$$J = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

J : le coût
$\hat{y}$: la prédiction
y : l'objectif
m : le nombre de valeurs total

Hypothèse et Précision

Comment évaluer les performances d'un algorithme ?

- Rappel : $\hat{y} = h(X)$
- Pour l'instant $h(X)$ pourrait être n'importe quelle hypothèse !
- C'est simplement une fonction qui prend des variables en *input* et qui sort un chiffre en *output*.

$$J = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

J	:	le coût
$\hat{y}$	:	la prédiction
y	:	l'objectif
m	:	le nombre de valeurs total

Hypothèse et Précision

Exemple

$$J = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

# de chambres	Surface m^2	Année Construction	# de Cheminées	Prix = y
3	8450	2003	0	208500
3	9600	1976	1	181500
3	11250	2001	1	223500
3	9550	1915	1	140000
4	14260	2000	1	250000
1	14115	1993	0	143000
3	10084	2004	1	307000
3	10382	1973	2	200000
2	6120	1931	2	129900
2	7420	1939	2	118000
3	11200	1965	0	129500
4	11924	2005	2	345000
2	12968	1962	0	144000
3	10652	2006	1	279500
2	10920	1960	1	157000
2	6120	1929	0	132000

2,69	10313,44	1972,63	0,94	193025,00	*moyenne*
3,00	10517,00	1974,50	1,00	169250,00	*mediane*

1 $h(X) = \hat{y}$	1 y	2 $(\hat{y} - y)^2$
0,00	208500,00	43472250000,00
0,00	181500,00	32942250000,00
0,00	223500,00	49952250000,00
0,00	140000,00	19600000000,00
0,00	250000,00	62500000000,00
0,00	143000,00	20449000000,00
0,00	307000,00	94249000000,00
0,00	200000,00	40000000000,00
0,00	129900,00	16874010000,00
0,00	118000,00	13924000000,00
0,00	129500,00	16770250000,00
0,00	345000,00	119025000000,00
0,00	144000,00	20736000000,00
0,00	279500,00	78120250000,00
0,00	157000,00	24649000000,00
0,00	132000,00	17424000000,00

3 $\sum(\hat{y} - y)^2$
6,71E+11

J
4,19E+10

4

Hypothèse et Précision

Exemple

J
1,01E+09

$$J = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

# de chambres	Surface m^2	Année Construction	Cheminée
3	8450	2003	0
3	9600	1976	1
3	11250	2001	1
3	9550	1915	1
4	14260	2000	1
1	14115	1993	0
3	10084	2004	1
3	10382	1973	2
2	6120	1931	2
2	7420	1939	2
3	11200	1965	0
4	11924	2005	2
2	12968	1962	0
3	10652	2006	1
2	10920	1960	1
2	6120	1929	0

$h(X) = \hat{y}$	y	$(\hat{y} - y)^2$
206320,00	208500,00	4,75E+06
209390,00	181500,00	7,78E+08
220140,00	223500,00	1,13E+07
203040,00	140000,00	3,97E+09
298090,00	250000,00	2,31E+09
107645,00	143000,00	1,25E+09
214610,00	307000,00	8,54E+09
213020,00	200000,00	1,70E+08
124510,00	129900,00	2,91E+07
131810,00	118000,00	1,91E+08
216270,00	129500,00	7,53E+09
286930,00	345000,00	3,37E+09
161810,00	144000,00	3,17E+08
217650,00	279500,00	3,83E+09
151390,00	157000,00	3,15E+07
124270,00	132000,00	5,98E+07

2,69	10313,44	1972,63	0,94	193025,00	*moyenne*
3,00	10517,00	1974,50	1,00	169250,00	*mediane*

$\sum(\hat{y} - y)^2$
3,24E+10

h(x) = 0	Room
2,1E+10	**1,16E+09**
h(x) = mean	R +Surface
2,32E+09	**1,09E+09**
h(x) = median	R+S+Année C
2,61E+09	
h(x) = rand()	R+S+Ac+Cheminée
6,08E+09	

Hypothèse et Précision

Création d'une hypothèse de régression **linéaire**

- Une régression linéaire a plusieurs variables.
- Il faut assigner un poids à chaque variable.

# de chambres	Surface m^2	Année Construction	Cheminée	Prix
x_1	x_2	x_3	x_4	y
3	8450	2003	0	208500

$$h_\theta(X) = \theta_0 + \theta_1 * 3 + \theta_2 * 8450 + \theta_3 * 2003 + \theta_4 * 0$$

- Pour *n* variables : $h_\theta(X) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$

Cost Function

La Régression Logistique

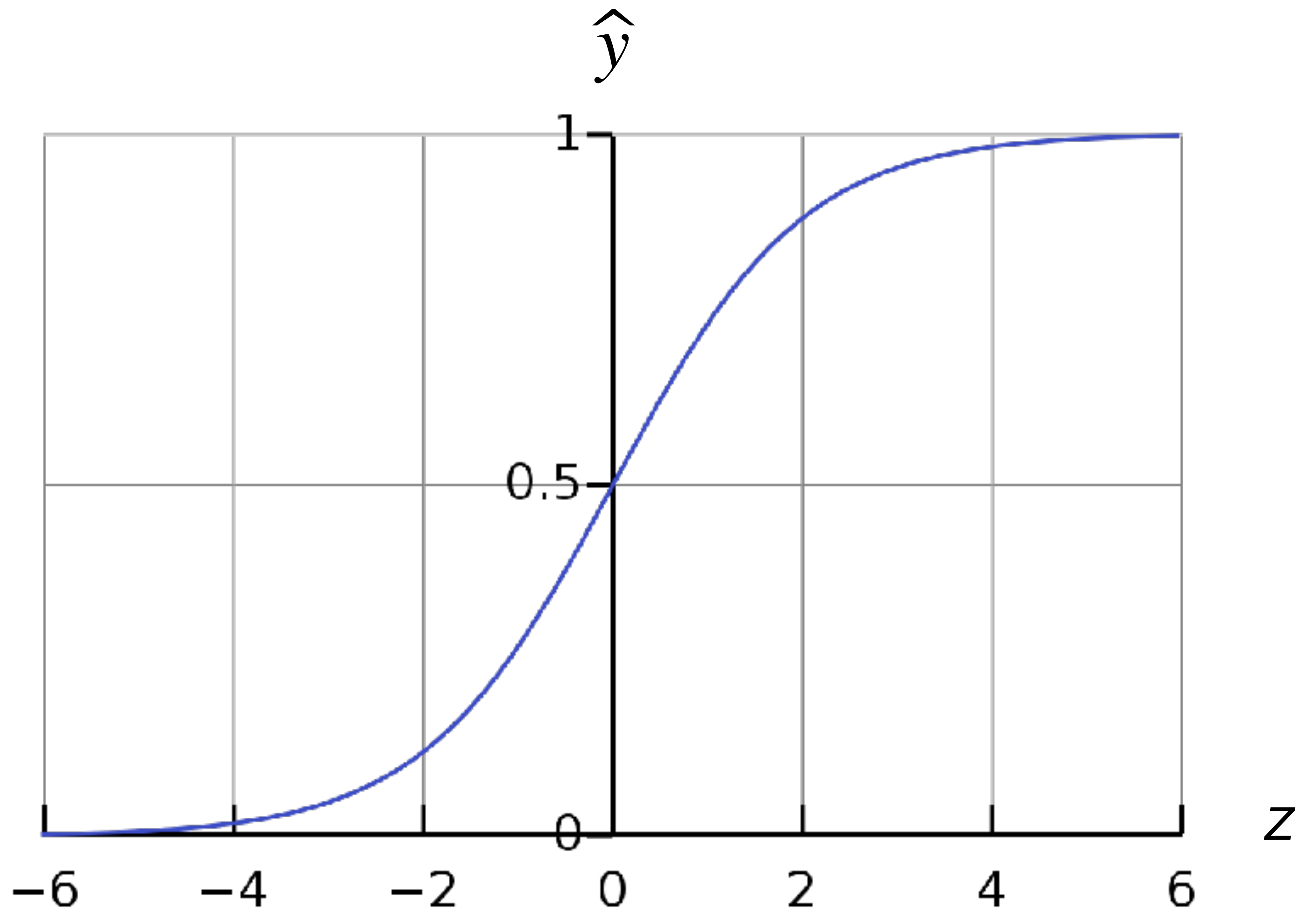
Hypothèse et Précision

Création d'une hypothèse de régression **logistique**

- La classification n'est pas un problème linéaire.
- Fonction logistique :

$(z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n)$

$$h_\theta(z) = \frac{e^z}{1 + e^z}$$



Hypothèse et Précision

Création d'une fonction de coût pour la régression **logistique**

- *y* est soit **1** soit **0** donc on ne peut pas calculer de distance.
- Le *coût* sera représenté par la confiance de *l'hypothèse* dans sa *prédiction*.
- Ex: Si $h_\theta(X) = 0.51$, le résultat sera interprété comme un **1** mais il est raisonnable d'y lire que la *prédiction* n'est pas '**confiante**'.

Pour *y* = 1 :

$$cost(h_\theta(X)) = -\log(h_\theta(x))$$

Pour *y* = 0 :

$$cost(h_\theta(X)) = -\log(1 - h_\theta(x))$$

Hypothèse et Précision

Création d'une fonction de coût pour la régression **logistique**

- Version finale

$$J = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

J : le coût
y : l'objectif

Pour $y = 1$:

$$1 - y = 0$$

Donc :

$$J = \log(h_\theta(x))$$

Pour $y = 0$:

$$-y = 0$$

Donc :

$$J = \log(1 - h_\theta(x))$$

Hypothèse et Précision

Création d'une fonction de coût pour la régression **logistique**

- Version finale

$$J = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

J : le coût
y : l'objectif

Date du match	Pluie?	Ecart avec le dernier match	...	Bléssé? = y
				0
				1
				1
				1

Hypothèse et Précision

Création d'une fonction de coût pour la régression **logistique**

- Encore une fois, il faut moyenner sur les *m* valeurs présentes dans le jeu de données.

- Version finale :

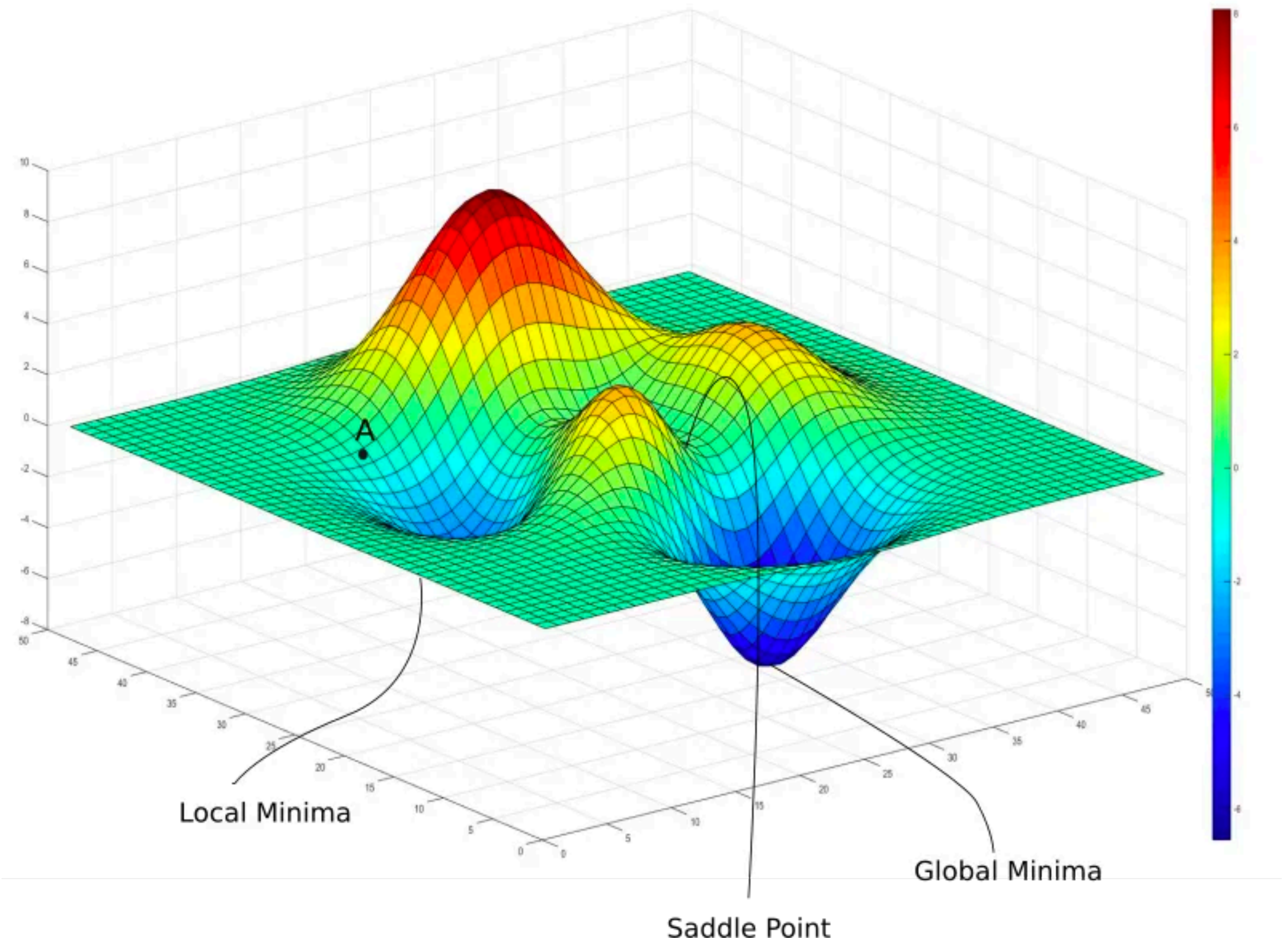
$$J = -\frac{1}{m}\sum_{i=1}^{m} -y^i \log(h_\theta(x^i)) - (1-y^i)\log(1-h_\theta(x^i))$$

J : le coût
y : l'objectif
m : le nombre de valeurs

Gradient Descent

Trouver le minimum d'une fonction

- On veut trouver la combinaison des paramètres θ_i qui réduirait un maximum la fonction de coût.



Trouver le minimum d'une fonction

Exemple en régression linéaire sur un cas

- Trouver les valeurs de $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4$ qui réduiraient le coût de ***J***.

$$h_\theta(x^1) = \theta_0 + \theta_1 * 3 + \theta_2 * 8450 + \theta_3 * 2003 + \theta_4 * 0$$

$$J = (h_\theta(x^1) - 208500)^2$$

<u>1ère solution</u> : *Générer la fonction pour tous les points et trouver le minimum local.*

➡ *Trop coûteux computationellement et en capacité de mémoire.*

➡ *Se rappeler des exemples en théorie des graphes.*

Trouver le minimum d'une fonction

Exemple en régression linéaire sur un cas

- Trouver les valeurs de $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4$ qui réduiraient le coût de ***J***.

$$h_\theta(x^1) = \theta_0 + \theta_1 * 3 + \theta_2 * 8450 + \theta_3 * 2003 + \theta_4 * 0$$

$$J = (h_\theta(x^1) - 208500)^2$$

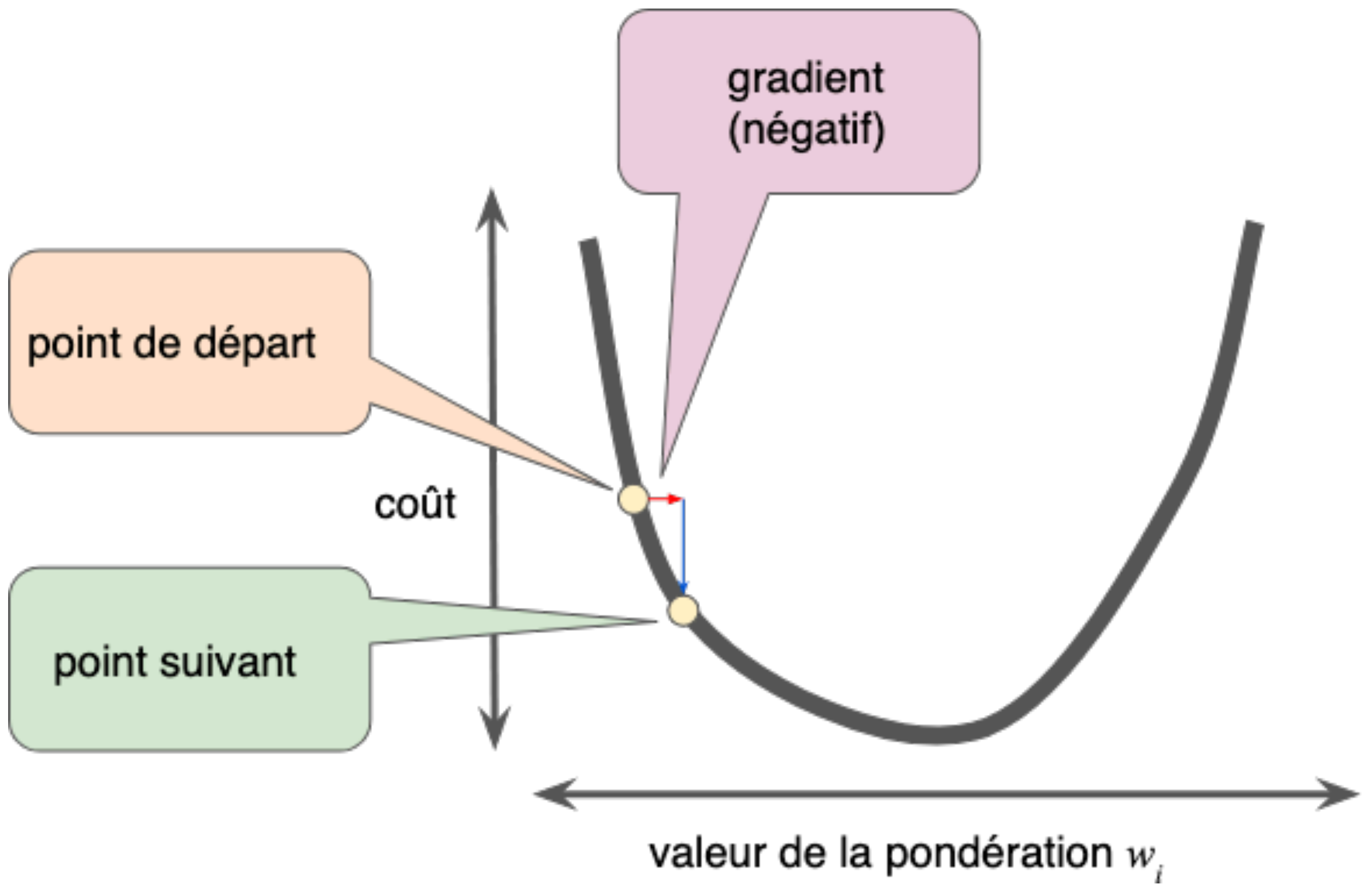
<u>2ème solution</u> : Prendre un point aléatoire et suivre le gradient vers le bas de façon itérative : ***Gradient Descent !***

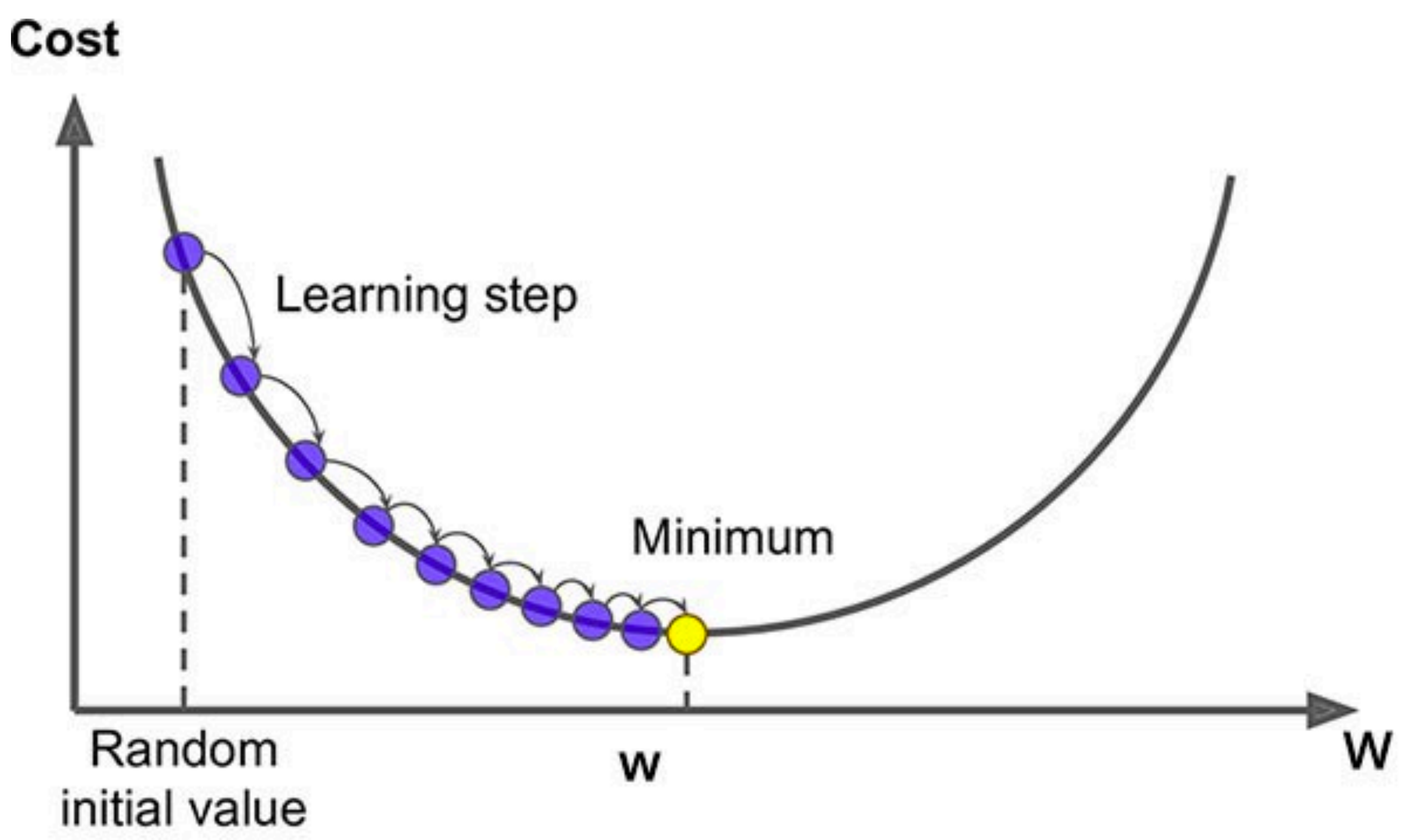
➡ *Cela demande juste de réaliser une dérivée partielle et locale, puis de mettre à jour les poids θ.*

Trouver le minimum d'une fonction

Exemple en régression linéaire sur un cas

- α est un paramètre que l'on nomme le '*Learning Step*', qui est la taille du saut de la nouvelle valeur de θ.
- α est un hyperparamètre sensible et important car il est au coeur de l'apprentissage de beaucoup de modèles et d'algorithmes en ML.





Trouver le minimum d'une fonction

Exemple en régression linéaire sur un cas

$$h_\theta(x^1) = \theta_0 + \theta_1 * 3 + \theta_2 * 8450 + \theta_3 * 2003 + \theta_4 * 0$$

$$J = (208500 - h_\theta(x^1))^2$$

Pour chaque θ

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j}$$

Nouvelle valeur de θ_j

Ancienne valeur de θ_j

Learning step

Gradient descent

Et on répète jusqu'à convergence (quand on atteint un minimum et que donc $\frac{\partial J}{\partial \theta_j} \approx 0$) .

Trouver le minimum d'une fonction

Exemple en régression linéaire sur tout les cas

- On applique le *gradient descent* sur chaque valeur dans le jeu de données pour trouver des poids moyens pour chaque θ.

$$J = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j}$$

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 * x_j^i$$