Université Constantine 2
جامعة قسنطينة 2

# Apprentissage machine 1

## Chapitre 2 : La régression linéaire

**Ouadfel Salima**
Faculté NTIC/IFA
salima.ouadfel@univ-constantine2.dz

Université Constantine 2
جامعة قسنطينة 2

# Apprentissage machine 1

# Chapitre 2 : La régression linéaire

Faculté NTIC/IFA

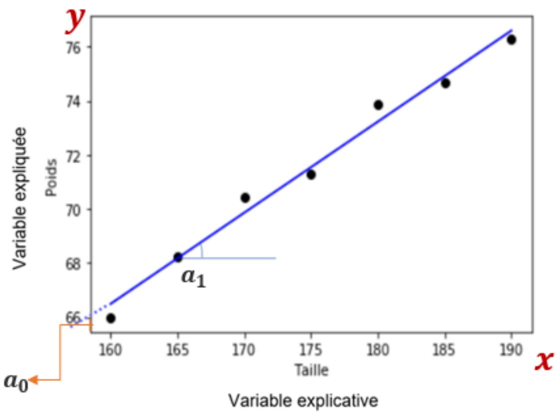
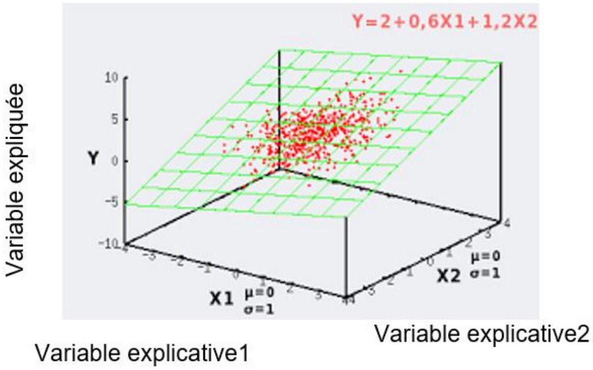
alima.ouadfel@univ-constantine2.dz

## Etudiants concernés

| Faculté/Institut | Département | Niveau | Spécialité |
|---|---|---|---|
| Nouvelles technologies | IFA | Master1 | STIC |

# La régression linéaire

-Le modèle de régression linéaire est un modèle d'apprentissage supervisé qui a pour objectif de trouver un modèle linéaire qui explique une variable expliquée $Y$ en fonction des variables explicatives indépendantes $X$.

| Régression linéaire simple | Régression lineaire multiple |
|---|---|
| Le modèle prédit une **variable expliquée Y** à partir **d'une variable explicative X**   | Le modèle prédit une **variable expliquée Y** à partir **de plusieurs variables explicatives X**   |

## Le modèle de régression linéaire

**Modèle :** $$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots. + a_px_p + \varepsilon$$

**Ecriture matricielle:**

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ & & . & \\ & & . & \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ a_p \end{pmatrix}}_{a} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_n \end{pmatrix}}_{\varepsilon}$$

$$Y = Xa + \varepsilon$$

**Paramètres:** $a = (a_0, a_1, a_2, \dots. a_p)$

**Fonction coût:** $$J(a) = \frac{1}{n}\sum_i^n \varepsilon_i^2$$

**But: minimiser:** $J(a)$

## Le modèle de régression linéaire

La méthode des moindre carrés est une méthode analytique qui permet de trouver le vecteur $\hat{a}$ qui minimise la fonction coût $\boldsymbol{J(a)} = \boldsymbol{\frac{1}{n}\sum_i^n \varepsilon_i^2}$

$$\hat{a} = (X^T X)^{-1} X^T Y$$

Mais elle devient difficile avec un très grand nombre de données et de caractéristiques à cause de l'inversion de la matrice $X^T X$.

Une alternative à la méthode des moindres carrés est une méthode approximative : l'algorithme ***Descente de Gradient*** pour trouver une solution approximative $\hat{a}$ .

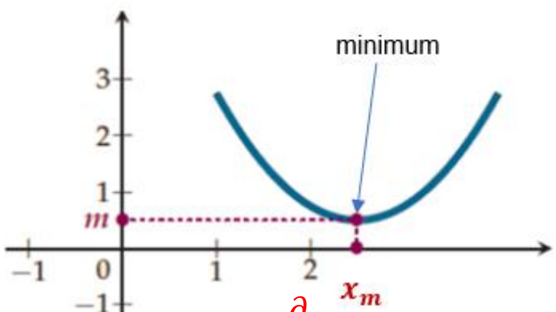
# La régression linéaire

## L'algorithme de descente du gradient

L'algorithme de la Descente de Gradient est un algorithme d'optimisation qui permet de trouver le minimum d'une fonction.

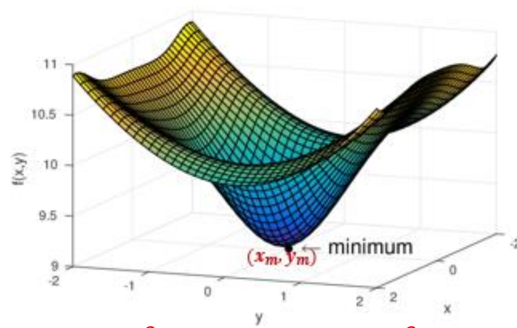
Entrée : une fonction $f: R^n \rightarrow R$

Objectif: trouver $\boldsymbol{x_m}$ tel que $f(\boldsymbol{x_m})$ est minimum




$$\frac{\partial}{\partial x} f(\boldsymbol{x_m}) = 0$$

avant $\boldsymbol{x_m}$ la dérivée est négative
($f$ est décroissante)
après $\boldsymbol{x_m}$ la dérivée est positive
($f$ est croissante)

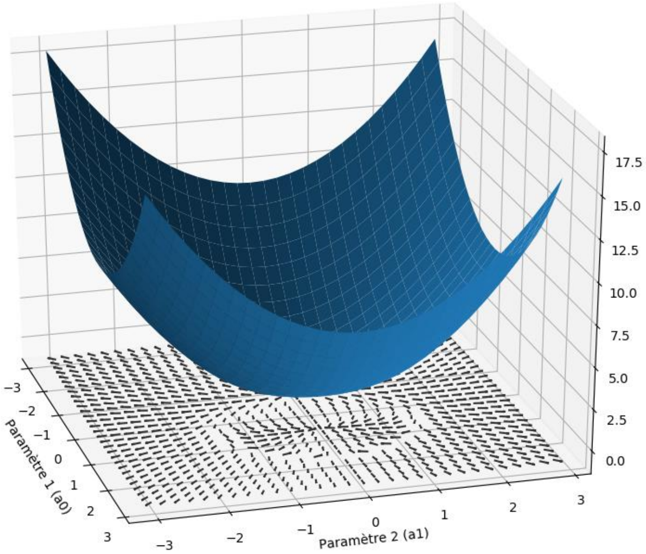


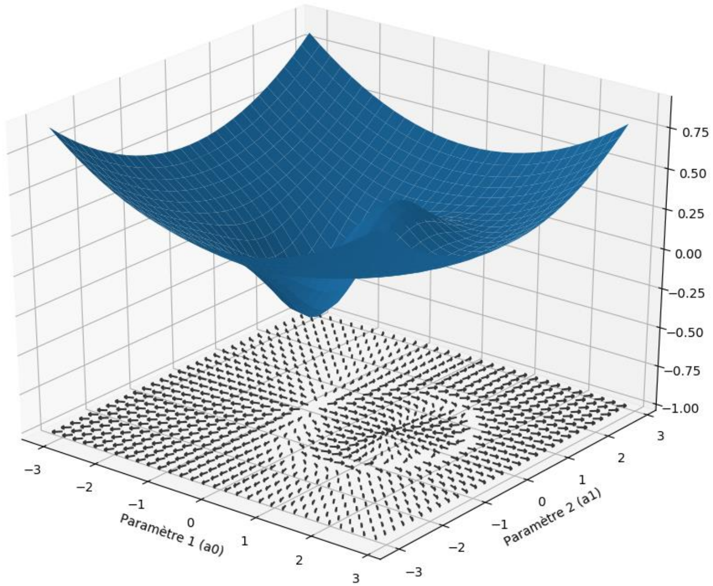

$\frac{\partial}{\partial x} f(\boldsymbol{x_m}) = 0$ et $\frac{\partial}{\partial y} f(\boldsymbol{y_m}) = 0$

avant $(\boldsymbol{x_m}, \boldsymbol{y_m})$ la dérivée est négative
($f$ est décroissante)
après $(\boldsymbol{x_m}, \boldsymbol{y_m})$ la dérivée est positive
($f$ est croissante)

## L'algorithme de descente du gradient

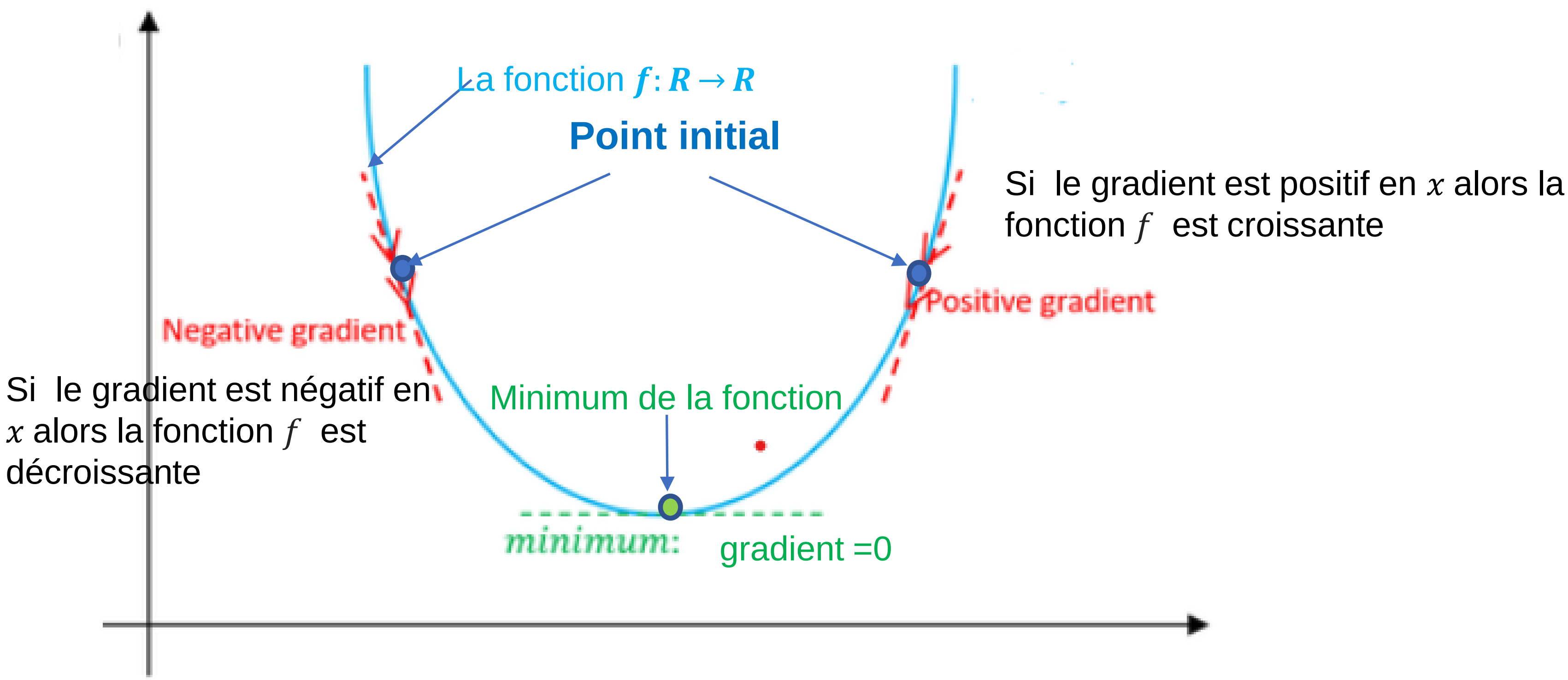
L'algorithme de descente du gradient est un algorithme itératif. A partir d'une point de la fonction (à minimiser) choisi arbitrairement, une suite de déplacement est effectuée dans la direction *opposée* au gradient, de manière à faire décroître la fonction.



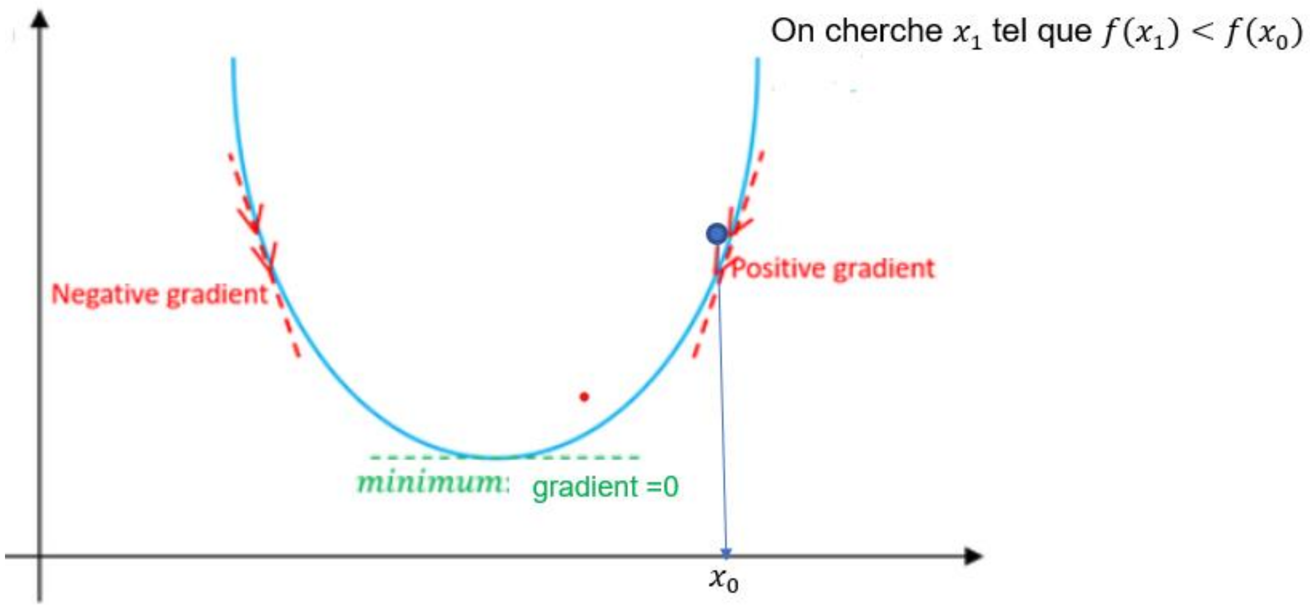
# L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



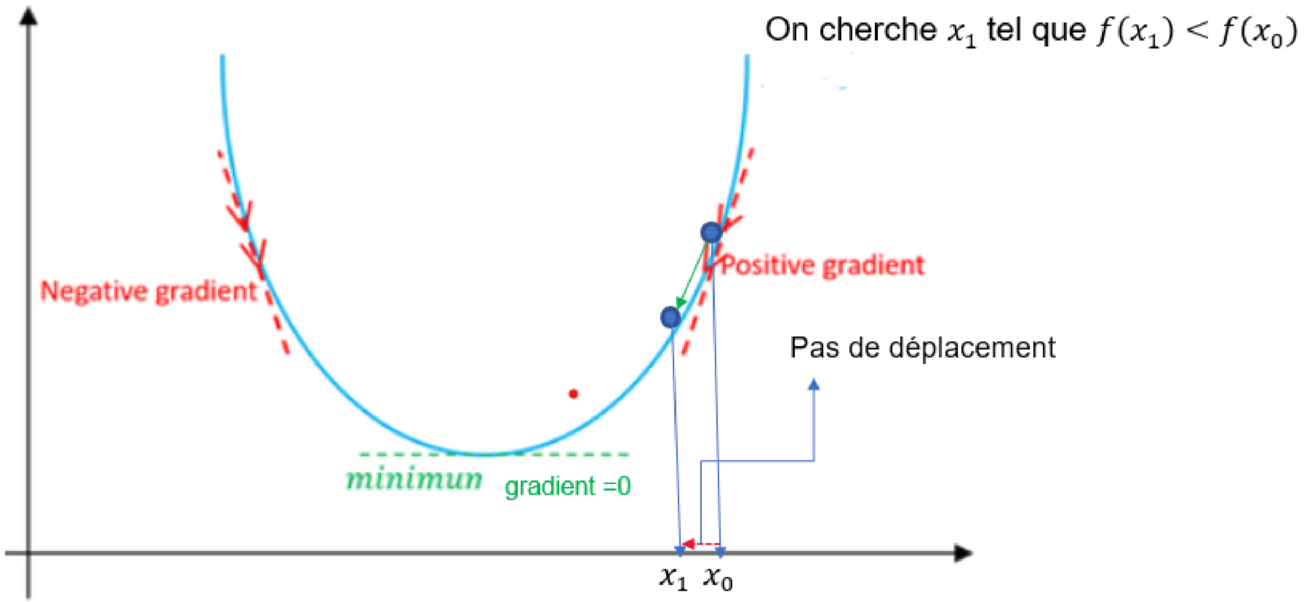
## L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



## L'algorithme de descente du gradient

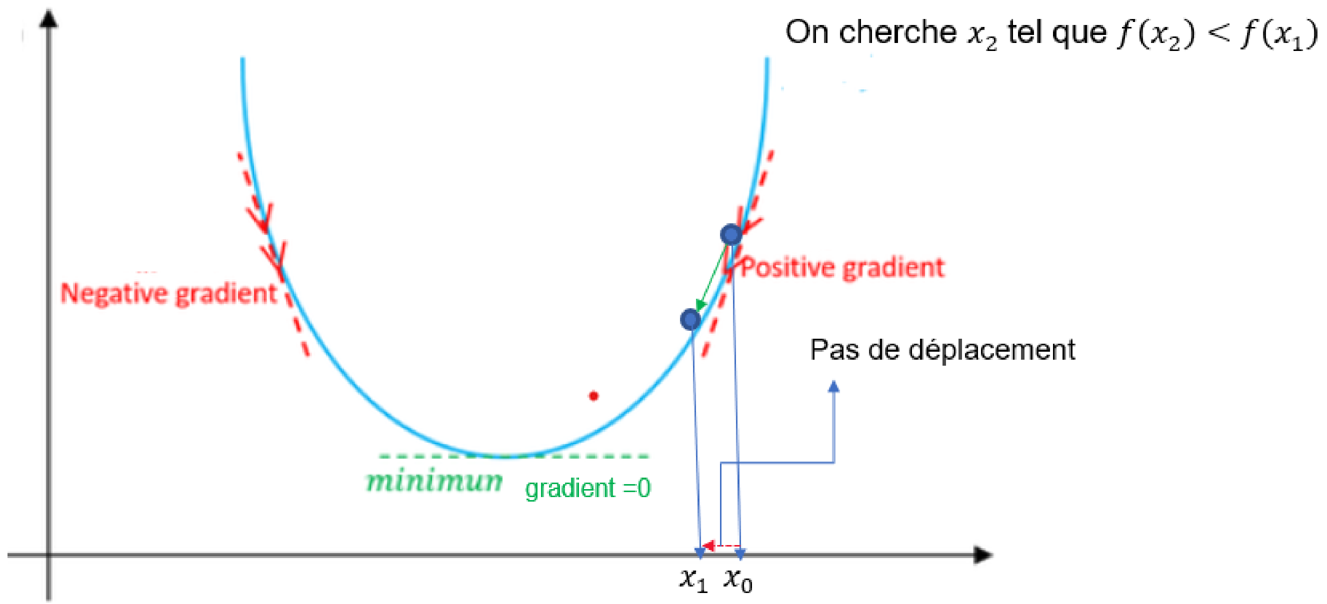
1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence




$$x_1 = x_0 - \frac{\partial}{\partial x} f(x_0)$$

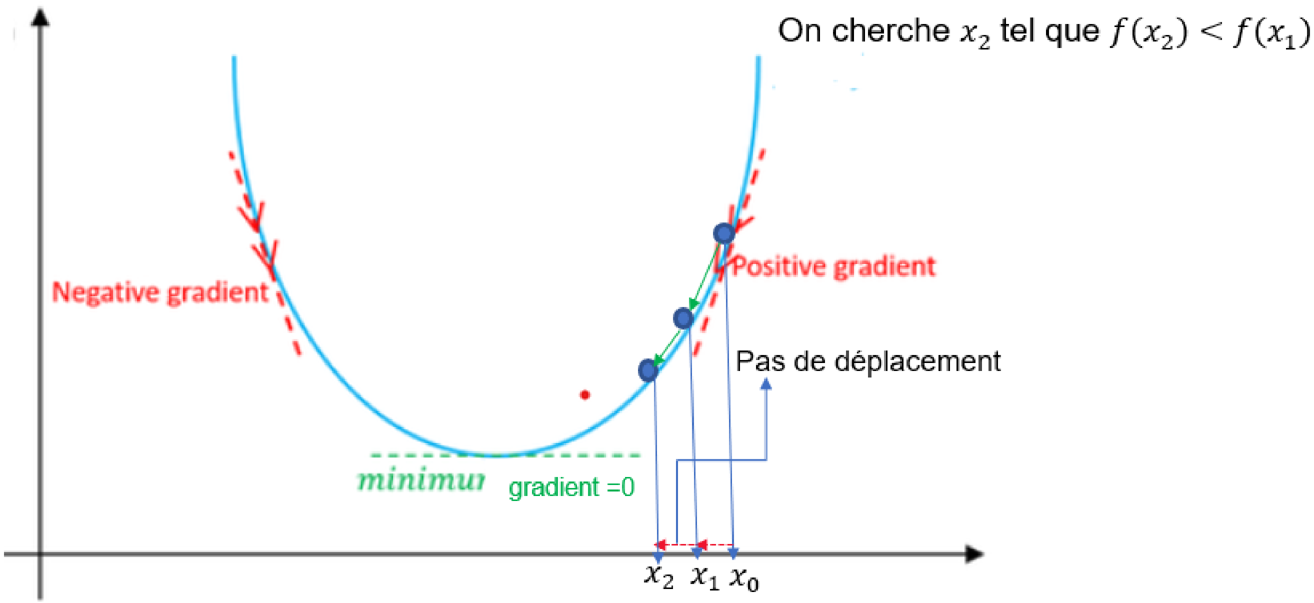
# L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



## L'algorithme de descente du gradient

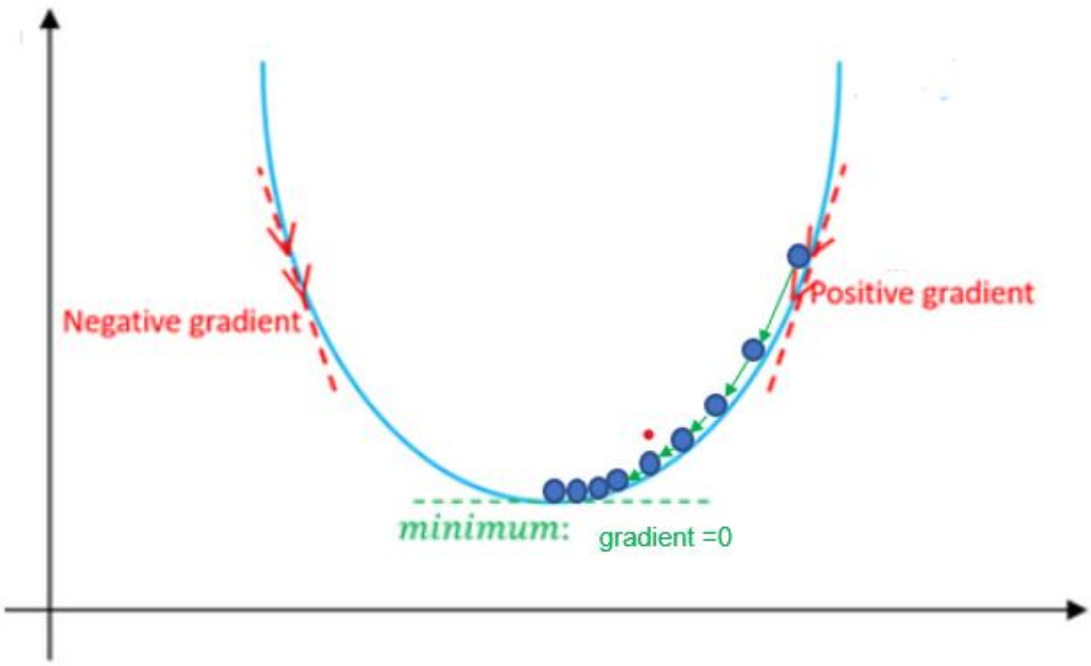
1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence




$$x_2 = x_1 - \frac{\partial}{\partial x} f(x_1)$$

## L'algorithme de descente du gradient

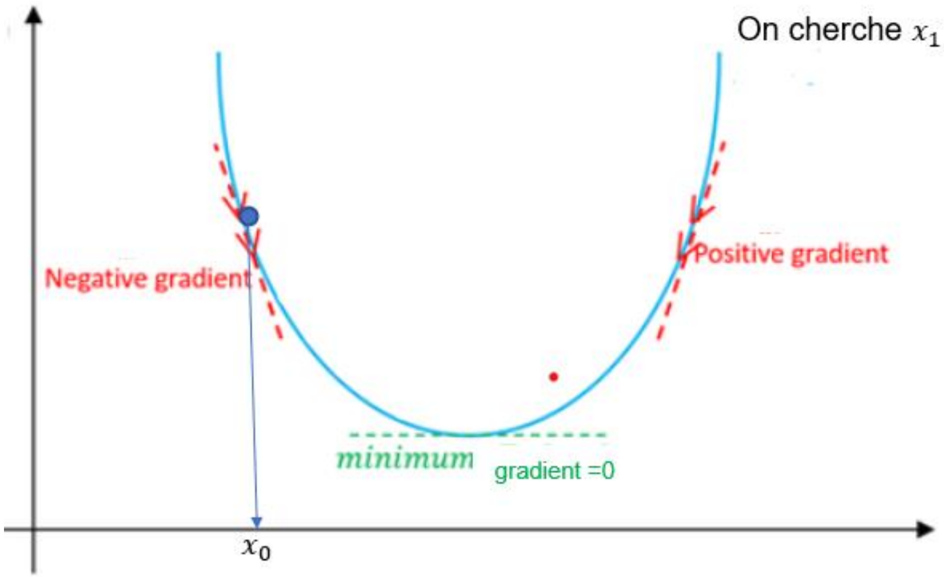
1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



## L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence

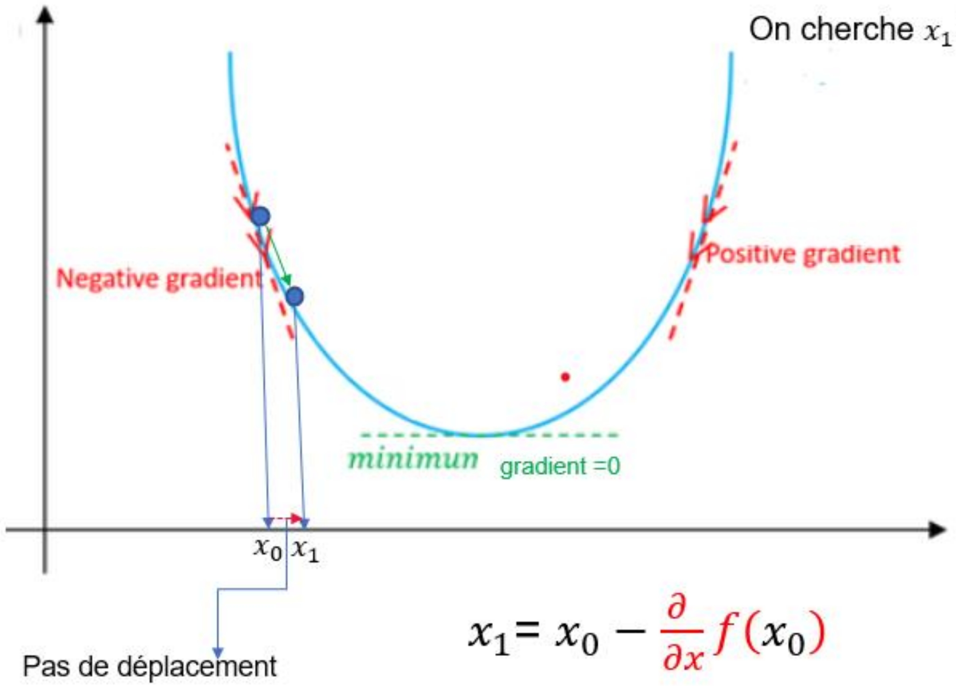
On cherche $x_1$ tel que $f(x_1) < f(x_0)$



# L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence

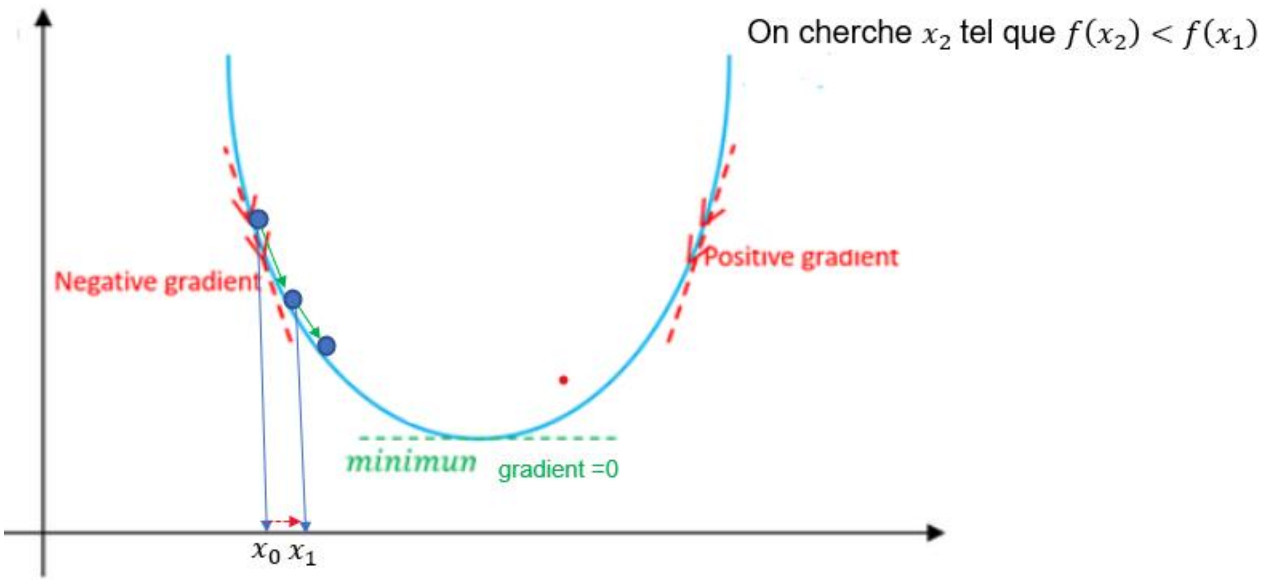
On cherche $x_1$ tel que $f(x_1) < f(x_0)$




$$x_1 = x_0 - \frac{\partial}{\partial x} f(x_0)$$

# L'algorithme de descente du gradient

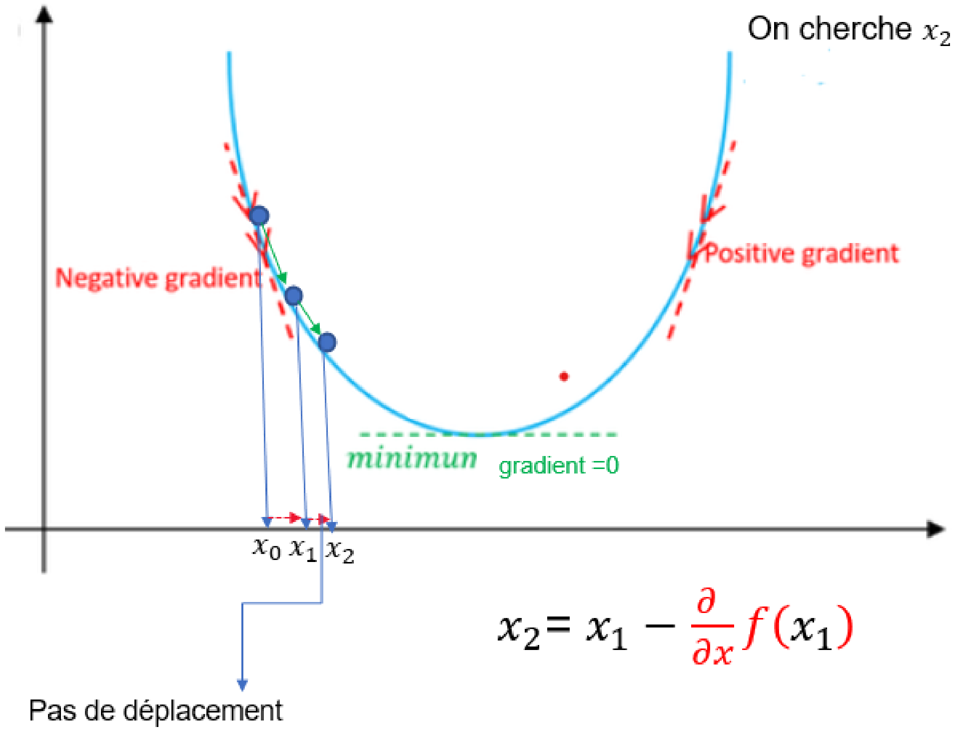
1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



# L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence

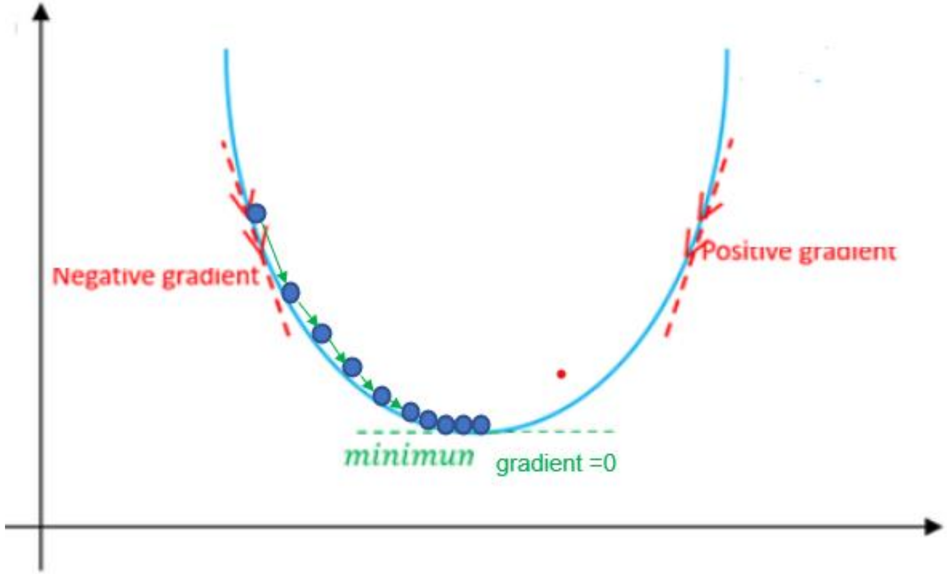
On cherche $x_2$ tel que $f(x_2) < f(x_1)$




$$x_2 = x_1 - \frac{\partial}{\partial x} f(x_1)$$

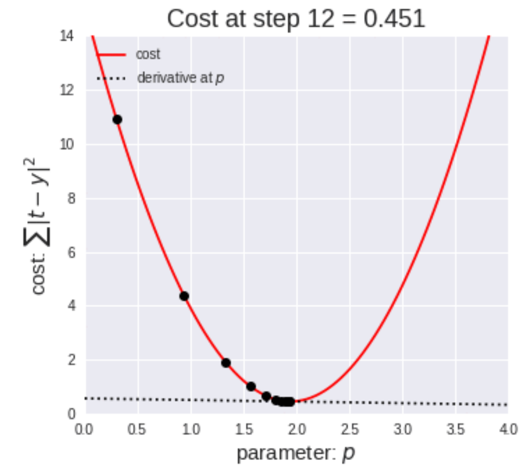
## L'algorithme de descente du gradient

1- On choisit un point de départ de coordonnée $x$ de la fonction $f$
2- On calcul le gradient en ce point
3- On se déplace dans la direction opposée au gradient
4- Revenir à 2 jusqu'à convergence



# L'algorithme de descente du gradient

Entrée: $f(x)$
Sortie : $argmin\ f(x)$




Etapes
1- initialiser l'algorithme avec $x^0$ choisi aléatoirement
2- iter=1 // initialiser le compter des itérations
3- Repeter
*// faire des changements sur $x^{iter-1}$dans le but de réduire $f(x)$*

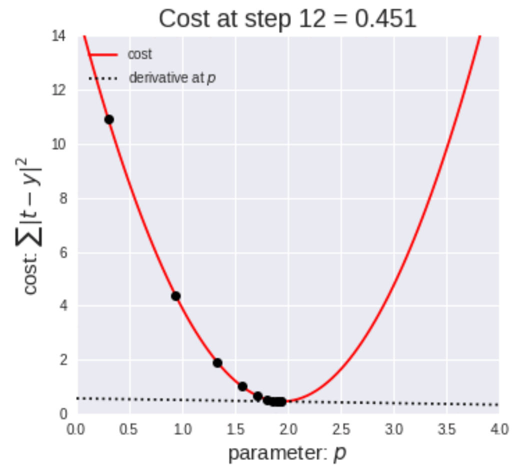
$$x^{iter} = x^{iter-1} - \boxed{\frac{\partial}{\partial x} f\left(x^{iter-1}\right)}$$

Dérivée de la fonction $f$
par rapport à $x$
au point $\left(x^{iter-1}\right)$

$iter = iter + 1$
Jusqu'à convergence

# L'algorithme de descente du gradient

Afin de contrôler le pas de déplacements, un hyperparamètre $\propto$ appelé **pas d'apprentissage** est introduit dans l'équation de mise à jour des valeurs du parameter $x$




Entrée: $f(x)$
Sortie : $argmin\ f(x)$

Etapes
1- initialiser l'algorithme avec $x^0$ choisi aléatoirement
2- iter=1 // initialiser le compter des itérations
3- Repeter
*// faire des changements sur $x^{iter-1}$ dans le but de réduire $f(x)$*

$$x^{iter} = x^{iter-1} - \boxed{\propto} * \boxed{\frac{\partial}{\partial x} f\left(x^{iter-1}\right)}$$

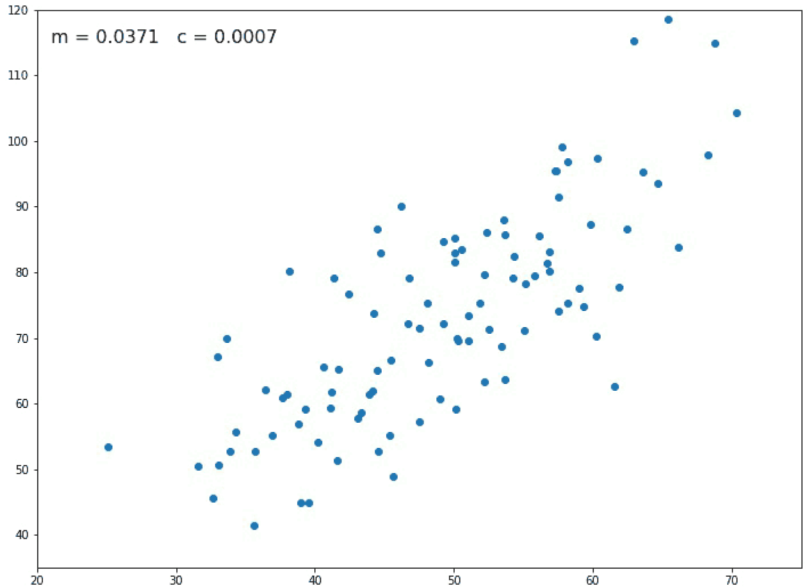
Pas d'apprentissage

Dérivée de la fonction $f$ par rapport à $x$ au point $\left(x^{iter-1}\right)$

$iter = iter + 1$
Jusqu'à convergence

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

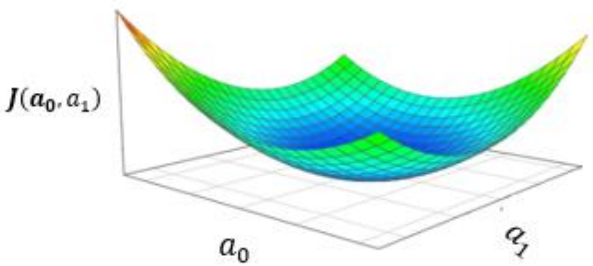



Modèle : $y = a_0 + a_1 x + \varepsilon$

Paramètres: $a_0$ *et* $a_1$

Fonction coût: $J(a_0, a_1) = \frac{1}{n} \sum_i^n \varepsilon_i^2$

But: minimiser $J(a_0, a_1)$

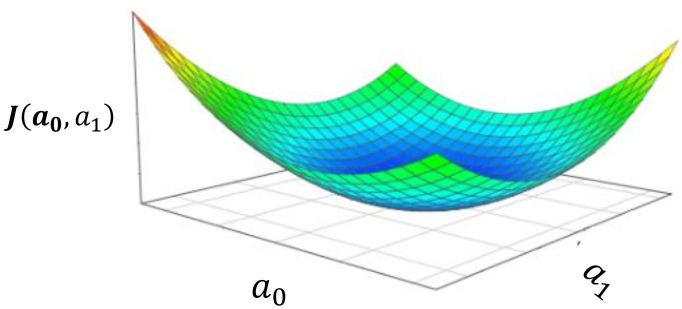


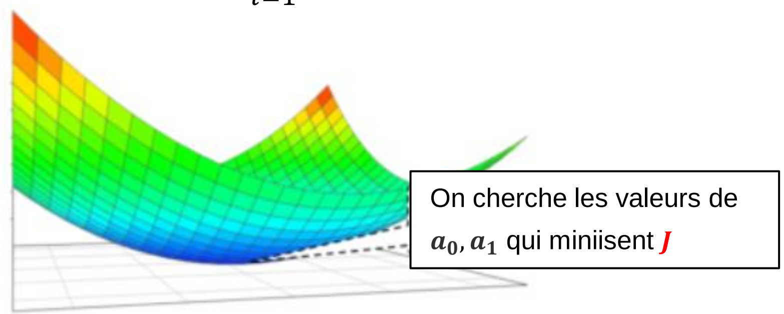
# La regression linéaire

## L’algorithme de descente du gradient

### Modèle de régression linéaire simple

$$J(a_0, a_1) = argmin\ \frac{1}{2n}\sum_{i=1}^{n}\varepsilon_i^2 = argmin\frac{1}{2n}\sum_{i=1}^{n}(y_i - (a_0 + a_1 x_i))^2$$



## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

Entrée: $J(\boldsymbol{a_0}, \boldsymbol{a_1}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - (\boldsymbol{a_0}+\boldsymbol{a_1}x_i))^2$

Sortie : $\underset{\boldsymbol{a_0},\boldsymbol{a_1}}{argmin}\, J(\boldsymbol{a_0}, \boldsymbol{a_1})$

Etapes

1- initialiser l'algorithme avec $a_0, a_1$ (par exemple $a_0^0 = 0, a_1^0 = 0$)

2- iter=1 // initialiser le compter des itérations

3- Repeter

*// faire des changements sur $a_0^{iter-1}$, $a_1^{iter-1}$ dans le but de réduire $J(\boldsymbol{a_0}, \boldsymbol{a_1})$*

$$a_0^{iter} = a_0^{iter-1} - \boxed{\propto} * \boxed{\frac{\partial}{\partial \boldsymbol{a_0}} J\left(a_0^{iter-1}, a_1^{iter-1}\right)}$$

Dérivée de la fonction $J$ par rapport à $\boldsymbol{a_0}$ au point $\left(a_0^{iter-1}, a_1^{iter-1}\right)$

Pas d'apprentissage

$$a_1^{iter} = a_1^{iter-1} - \boxed{\propto} * \boxed{\frac{\partial}{\partial \boldsymbol{a_1}} J\left(a_0^{iter-1}, a_1^{iter-1}\right)}$$

Dérivée de la fonction $J$ par rapport à $\boldsymbol{a_1}$ au point $\left(a_0^{iter-1}, a_1^{iter-1}\right)$

$iter = iter + 1$

jusqu'à atteindre un nombre max d'itérations

# La regression linéaire

La dérivée partielle de J en $\boldsymbol{a_1}$:

$$\frac{\partial}{\partial \boldsymbol{a_1}} J(\boldsymbol{a_0}, \boldsymbol{a_1}) = \frac{\partial}{\partial \boldsymbol{a_1}} \left( \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\boldsymbol{a_1} x_i + \boldsymbol{a_0}))^2 \right)$$

$$\boxed{\frac{\partial}{\partial \boldsymbol{a_1}} \boldsymbol{J}(\boldsymbol{a_0}, \boldsymbol{a_1}) = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} -\boldsymbol{x_i}(\boldsymbol{y_i} - (\boldsymbol{a_1} \boldsymbol{x_i} + \boldsymbol{a_0}))}$$

La dérivée partielle de J en $\boldsymbol{a_0}$:

$$\frac{\partial}{\partial \boldsymbol{a_0}} J(\boldsymbol{a_0}, \boldsymbol{a_1}) = \frac{\partial}{\partial \boldsymbol{a_0}} \left( \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\boldsymbol{a_1} x_i + \boldsymbol{a_0}))^2 \right)$$

$$\boxed{\frac{\partial}{\partial \boldsymbol{a_0}} \boldsymbol{J}(\boldsymbol{a_0}, \boldsymbol{a_1}) = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} -(\boldsymbol{y_i} - (\boldsymbol{a_1} \boldsymbol{x_i} + \boldsymbol{a_0}))}$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire multiple

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots . + a_p x_p + \varepsilon$$

$$y = a_0 x_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots . + a_p x_p + \varepsilon$$

**avec** $x_0 = 1$

$$J(a_0, a_1, \dots . a_p) = \frac{1}{2n} \sum_{i=1}^{n} \varepsilon_i^2$$

$$\boxed{J(a_0, a_1, \dots . a_p) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - (a_0 + a_1 x_{i1} + a_2 x_{i2} + a_3 x_{i3} + \cdots . + a_p x_{ip} )\right)^2}$$

# La regression linéaire

## L'algorithme de descente du gradient

### Modèle de régression linéaire multiple

Entrée: $J(a_0, a_1, \dots . a_p) =$

Sortie : $argmin J(a_0, a_1, \dots . a_p)$

Etapes

1- initialiser l'algorithme avec $a_j^0$, (par exemple $a_j^0 = 0, j = 0 \dots p$)

2- iter=1 // initialiser le compter des itérations

3- Repeter

*// faire des changements sur $a_j$ dans le but de réduire* $J(a_0, a_1, \dots . a_p)$

$$a_j^{iter} = a_j^{iter-1} - \boxed{\alpha} * \boxed{\frac{\partial}{\partial a_j} J\left(a_j^{iter-1}\right)}$$

Pas d'apprentissage

Dérivée de la fonction $J$ par rapport à $a_j$ au point $\left(a_j^{iter-1}\right)$

$iter = iter + 1$

jusqu'à atteindre un nombre max d'itérations

La dérivée partielle de J en $a_j \quad j = 0..p$

$$\frac{\partial}{\partial a_j} J\left(a_0, \ldots., a_p\right) = \frac{\partial}{\partial a_j}\left(\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - (a_0x_0 + a_1x_{i1} + a_2x_{i2} + \cdots . + a_px_{ip})\right)^2\right)$$

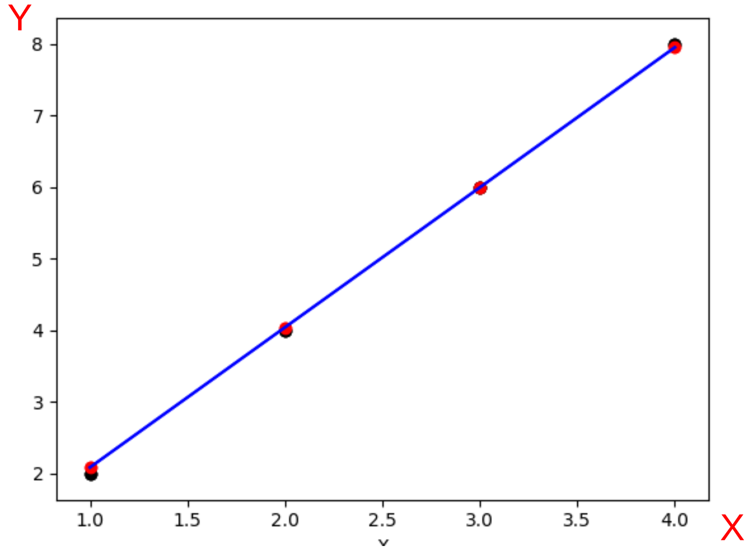
$$\boxed{\frac{\partial}{\partial a_j} J\left(a_0, \ldots., a_p\right) = \frac{1}{n}\sum_{i=1}^{n} -x_j\left(y_i - (a_0x_0 + a_1x_{i1} + a_2x_{i2} + \cdots . + a_px_{ip})\right)^2}$$

## L'algorithme de descente du gradient

**Modèle de régression linéaire simple**

### Exemple

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |



## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

### Exemple

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

Initialisation:

$a_1^0 = \mathbf{0}, a_0^0 = \mathbf{0}$

$\hat{\boldsymbol{y}} = \mathbf{0} + \mathbf{0} * \boldsymbol{x}$

$\propto = \mathbf{0.01}$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple**

Initialisation: $a_1^0 = \mathbf{0}, a_0^0 = \mathbf{0}$ $\quad \hat{y} = \mathbf{0} + \mathbf{0} * x \quad \propto = \mathbf{0,01}$

Iter= 1

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

**Calcul de la fonction coût**

$$J(a_0, a_1) = 1/2n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$J\left(a_0^{iter-1}, a_1^{iter-1}\right) = 1/8[(2-0)^2 + (4-0)^2 + (6-0)^2 + (8-0)^2] = 15$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple**

Initialisation: $a_1^0 = \mathbf{0}, a_0^0 = \mathbf{0}$ $\quad \hat{\boldsymbol{y}} = \mathbf{0} + \mathbf{0} * \boldsymbol{x}$ $\quad \propto = \mathbf{0,01}$

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

Iter= 1

**Calcul des dérivées partielles**

$$\frac{\partial}{\partial a_0} J(a_0^{iter-1}, a_1^{iter-1}) = \frac{1}{4}\sum_{i=1}^{4} -\left(y_i - (a_1^{iter-1} x_i + a_0^{iter-1})\right)$$

$$\frac{\partial}{\partial a_0} J\left(a_0^{iter-1}, a_1^{iter-1}\right) = 1/4\{-(2-(0+1*0)) \; -(4-(0+2*0)) \; -.....) = -5$$

$$\frac{\partial}{\partial a_1} J\left(a_0^{iter-1}, a_1^{iter-1}\right) = \frac{1}{4}\sum_{i=1}^{4} -x_i\left(y_i - (a_1^{iter-1} x_i + a_0^{iter-1})\right)$$

$$\frac{\partial}{\partial a_1} J\left(a_0^{iter-1}, a_1^{iter-1}\right) = 1/4\{-1*(2-(0+1*0)) \; -2*(4-(0+2*0)) \; -.....) = -15$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple**

Initialisation: $a_1^0 = \mathbf{0}, a_0^0 = \mathbf{0}$ $\quad \hat{\boldsymbol{y}} = \mathbf{0} + \mathbf{0} * \boldsymbol{x}$ $\quad \propto = \mathbf{0,01}$

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

Iter= 1

**Mise à jour des paramètres**

$$\mathrm{a}_1^{\mathrm{iter}} = \mathrm{a}_1^{\mathrm{iter}-1} - \propto * \frac{\partial}{\partial \mathrm{a}_1} \mathrm{J}\left(\mathrm{a}_1^{\mathrm{iter}-1}, \mathrm{a}_0^{\mathrm{iter}-1}\right) = 0\text{-}0.01 * -15 = 0.15$$

$$\mathrm{a}_0^{\mathrm{iter}} = \mathrm{a}_0^{\mathrm{iter}-1} - \propto * \frac{\partial}{\partial \mathrm{a}_0} \mathrm{J}\left(\mathrm{a}_1^{\mathrm{iter}-1}, \mathrm{a}_0^{\mathrm{iter}-1}\right) = 0\text{-}0.01*\text{-}5 = 0.05$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

### Exemple

$$a_1^1 = \mathbf{0.15}\ , a_0^1 = \mathbf{0.05}\quad \hat{\boldsymbol{y}} = 0.05 + \mathbf{0.15}\boldsymbol{x}$$

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

Iter= 2

**Calcul de la fonction coût**

$$J(a_0, a_1) = 1/2n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$J\left(a_0^{iter-1}, a_1^{iter-1}\right) = 1/8[(2-(0.05+0.15*1))^2 + (4-(0.05+0.15*2))^2 + (6-(0.05+0.15*3))^2 + \ (8-(0.05+0.15*4))^2] = 12,604$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple** $a_1^1 = \mathbf{0.15}$ , $a_0^1 = \mathbf{0.05}$ $\hat{\boldsymbol{y}} = 0.05 + \mathbf{0.15}\boldsymbol{x}$

Iter= 2

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

**Calcul des dérivées partielles**

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_0}}\boldsymbol{J}(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}) = \frac{1}{4}\sum_{i=1}^{4} -\left(y_i - (a_1^{iter-1}x_i + a_0^{iter-1})\right)$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_0}}\boldsymbol{J}\left(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}\right) = 1/4\{-(2-(0.05+0.15*1)) \quad -(4-(0.05+0.15*2)) \quad -.....)$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_0}}\boldsymbol{J}\left(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}\right) = -4.575$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_1}}\boldsymbol{J}\left(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}\right) = \frac{1}{4}\sum_{i=14} -x_i\left(y_i - (a_1^{iter-1}x_i + a_0^{iter-1})\right)$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_1}}\boldsymbol{J}\left(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}\right) = 1/4\{-1*(2-(0.05+1*0.15)) \quad -2*(4-(0.05+2*0.15)) \quad -.....)$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial a_1}}\boldsymbol{J}\left(\boldsymbol{a_0^{iter-1}}, \boldsymbol{a_1^{iter-1}}\right) = -13.75$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple** $a_1^1 = \mathbf{0.15}$ , $a_0^1 = \mathbf{0.05}$ $\hat{y} = 0.05 + 0.15x$

Iter= 2

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

**Mise à jour des paramètres**

$$\mathrm{a}_1^{\mathrm{iter}} = \mathrm{a}_1^{\mathrm{iter}-1} - \propto * \frac{\partial}{\partial \mathrm{a}_1} \mathrm{J}\left(\mathrm{a}_1^{\mathrm{iter}-1}, \mathrm{a}_0^{\mathrm{iter}-1}\right) = 0.15 - 0.01 * -13.75 = 0.2875$$

$$\mathrm{a}_0^{\mathrm{iter}} = \mathrm{a}_0^{\mathrm{iter}-1} - \propto * \frac{\partial}{\partial \mathrm{a}_0} \mathrm{J}\left(\mathrm{a}_1^{\mathrm{iter}-1}, \mathrm{a}_0^{\mathrm{iter}-1}\right) = 0.05 - 0.01 * -4.575 = 0.09575$$

## L'algorithme de descente du gradient

### Modèle de régression linéaire simple

**Exemple**

$$a_1^2 = \mathbf{0.2875}\,.,\, a_0^2 = \mathbf{0.09575}$$
$$\hat{y} = \mathbf{0.}09575 + \mathbf{0.2875}x$$

Iter= 3

| $X$ | $Y$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

**Calcul de la fonction coût**

**…..**

**Calcul des dérivées partielles**

**…….**

**Mise à jour des paramètres**

**…**

## L'algorithme de descente du gradient

### Mise à l'échelle des données

Pour accélérer la convergence de l'algorithme et ne pas atteindre la limite mathématique de l'ordinateur, on effectue une opération de mise à l'échelle.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

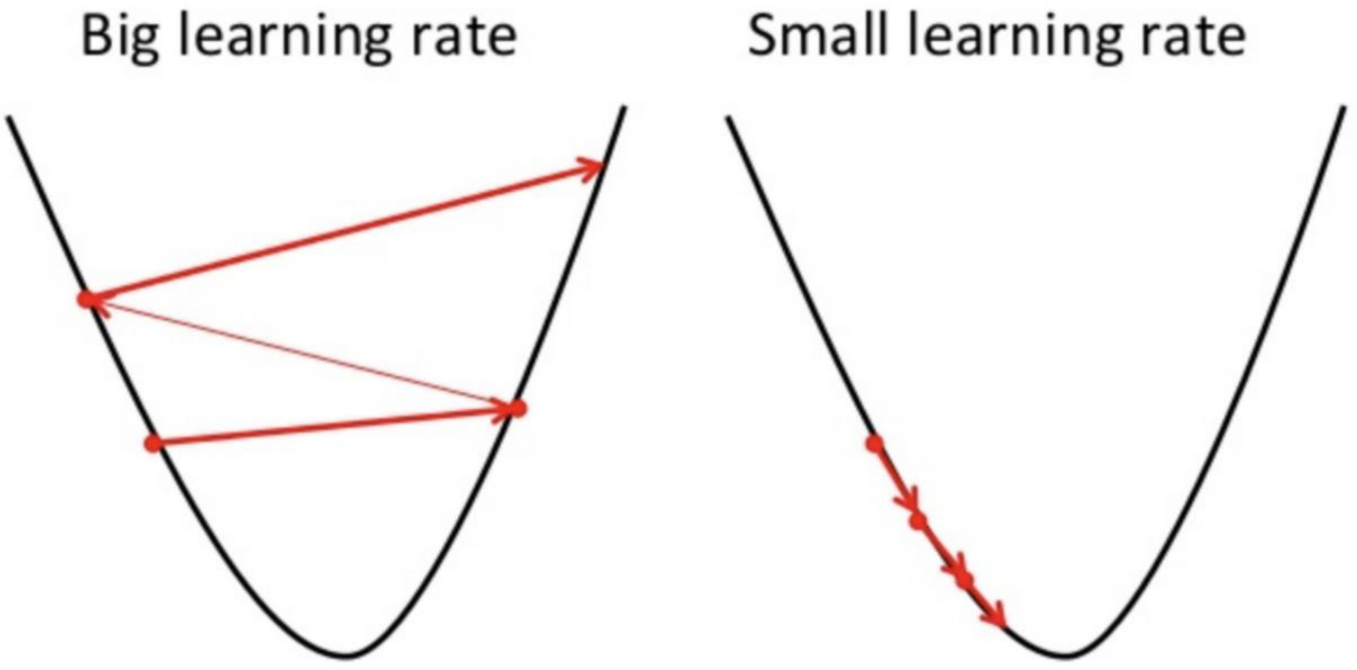
**Normalisation**

$$X' = \frac{X - \mu}{\sigma}$$

**Standardisation**

**L'algorithme de descente du gradient**

**Le pas d'apprentissage**



# La regression linéaire

## En résumé   Modèle de régression linéaire

**Modèle** $y = a_0 + \sum_{k=1}^{p} a_k x_k + \varepsilon$

**Coût** $$J(a_0, a_1, \dots . a_p) = 1/2n \sum_{i=1}^{n} \varepsilon_i^2 = 1/2n \sum_{i=1}^{n} \left( y_i - a_0 + \sum_{k=1}^{p} a_k x_{ik} + \varepsilon \right)^2$$

**Objectif** **minimiser** $J(a_0, a_1, \dots . a_p)$

**Méthode des moindres carrés**

Solution exacte
Aucun paramètre
Besoin de calculer

$$(X^T X)^{-1} X^T Y$$

Difficile si n et p est très grand
Inverse une matrice de (p+1)*(p+1)

**Algorithme de descente du gradient**

Solution approximative
Algorithme Itératif
Paramètre: le pas d'apprentissage
Nécessite de normaliser les données
Efficace si n est très grand
Calcul un vecteur (p+1)