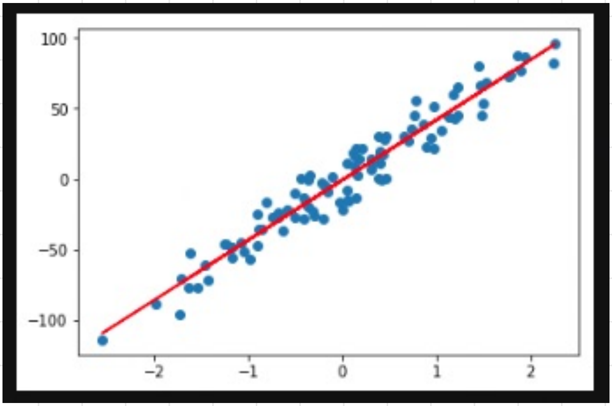
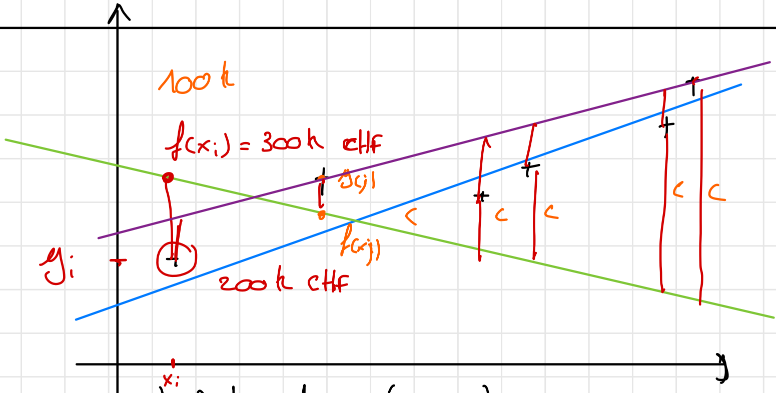
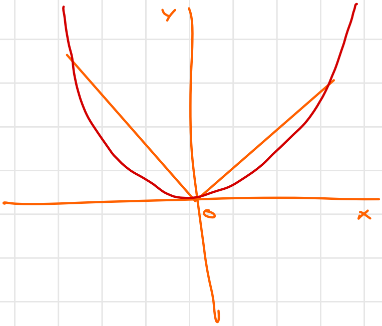
La regression lineaire est un modèle de machine learning simple. Mais il utilise les mêmes fondamentaux que les models plus complexe, qui permettent de faire de la reconnaissance vocale etc...



4 étapes :

1) Dataset
   y : target
   x : features
2) Modèle → prediction
   paramètres (tuning)
3) fonction de coûts (cost function) → evaluate model
4) algorithme de minimalisation
   minimalisation de la fonction de coûts





1) Dataset : (x, y) $m = 6$, $n = 1$

2) Model : $f(x) = ax + b$; a ; b are the parameters of my model
Initially we set a and b randoms

3) Cost function : measures the eross between the model and the data set

Made with Goodnotes

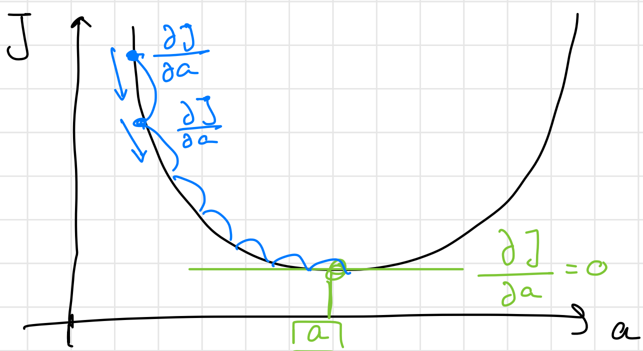
we can say ok It was 300 I predicted 200, so I have an erros of 100k

$$J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(\overline{f(x_i) - y_i}\right)^2 \longrightarrow \text{MSE}$$

euclidian distance or L2 norm

J is a function about parameters, because the parameters wills directlly influence the errors

4) learning algorithm (minimization of cost function)



cost function (in that case) is a sum of squared, thus J will looks like a squared function.

I want to find the min of J regarding a

to find a multiple possibilities :
- les moindres carrés
- gradient descent

Summary :

1. Dataset : $y \in \mathbb{R}^{m \times 1}$ and $x \in \mathbb{R}^{m \times n}$ where n is the nb of features

m is the no number of records

2. Model : $f(x) = ax + b$

3. cost function : $J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}(\underbrace{ax_i + b}_{f(x_i)} - \underbrace{y_i}_{y_i})^2$

4. minimization algo : gradient descent

$$a_{t+1} = a_t - \alpha \boxed{\frac{\partial J(a_t)}{\partial a}}$$

$<$

$>$

0

$$J(a,b) = \frac{1}{2m} \sum (ax + b - y)^2$$

$$f(a,b) = ax + b$$

$$g(f) = f^2$$

$$(g \circ f)' = f' \cdot g'(f)$$

$$\frac{\partial J}{\partial a} = \frac{1}{2m} \sum 2x (ax + b - y)$$

$$\frac{\partial J}{\partial b} = \frac{1}{2m} \sum (ax + b - y)$$

$$F = X . \Theta$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \dots \\ f(x_m) \end{bmatrix} \quad \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & | \\ x_n & 1 \end{bmatrix} \quad \begin{bmatrix} a \\ b \end{bmatrix}$$

$m \times 1$ $\qquad$ $m \times 2$ $(n+1)$ $\qquad$ 2×1 $(n+1)$

$$J(a,b) = \frac{1}{2m} \sum_{i=1}^{m} (\underbrace{ax_i + b}_{X\Theta} - \underbrace{y_i}_{Y})^2$$

$X\Theta$ $m \times 1$ $\quad$ Y $m \times 1$

$$\frac{1}{2m} \begin{bmatrix} -^2 \\ -^2 \\ -^2 \\ -^2 \end{bmatrix}^2$$

$$J(\theta) = \frac{1}{2m} \sum (X\theta - Y)^2$$

$$\begin{bmatrix} \dfrac{\partial J(a,b)}{\partial a} \\[2ex] \dfrac{\partial J(a,b)}{\partial b} \end{bmatrix} \qquad \underset{(n+1)\times 1}{\frac{\partial J(\theta)}{\partial \theta}} = \frac{1}{m} \underset{(n+1 \times m)}{\boxed{X^T}} \underset{m \times 1}{(X\theta - Y)}$$

$$X = \begin{bmatrix} x_i & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad m \times (n+1)$$

$$X^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & - & 1 \end{bmatrix}$$

$$(n+1) \times m$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \alpha \begin{bmatrix} \dfrac{\partial J}{\partial a} \\[2ex] \dfrac{\partial J}{\partial b} \end{bmatrix}$$

$$\downarrow$$

$$\underset{(n+1)\times 1}{\theta} = \underset{(n+1)\times 1}{\theta} - \alpha \underset{(n+1)\times 1}{\frac{\partial J}{\partial \theta}}$$

Made with Goodnotes