MINITEL

MACHINE LEARNING

02 REGRESSION WITH MULTIPLE INPUT VARIABLES

THÉO GACHET

UNE SÉRIE DE FORMATIONS

PARTIE 1 :
SUPERVISED MACHINE LEARNING : REGRESSION AND CLASSIFICATION

Introduction to Machine Learning
Regression with multiple input
Classification

PARTIE 2 :
ADVANCED LEARNING ALGORITHMS

Neural Networks
Neural Networks training
Advice for applying Machine Learning
Decision Trees

PARTIE 3 :
UNSUPERVISED LEARNING, RECOMMENDERS, REINFORCEMENT LEARNING

Unsupervised Learning
Recommander Systems
Reinforcement Learning

01 SUPERVISED MACHINE LEARNING : REGRESSION AND CLASSIFICATION

SUPERVISED VS. UNSUPERVISED ML

What is Machine Learning ?
Supervised Learning
Unsupervised Learning
Jupyter Notebooks
Lab : Python and Jupyter Notebooks

REGRESSION MODEL

Linear regression model
Lab : Model representation
Cost function formula
Cost function intuition
Visualizing the cost function
Visualization examples
Lab : Cost function

TRAIN THE MODEL WITH GRADIENT DESCENT

Gradient descent
Implementing gradient descent
Gradient descent intuition
Learning rate
Gradient descent for linear regression
Running gradient descent
Lab : Gradient descent

RAPPEL

Supervised learning

x → y

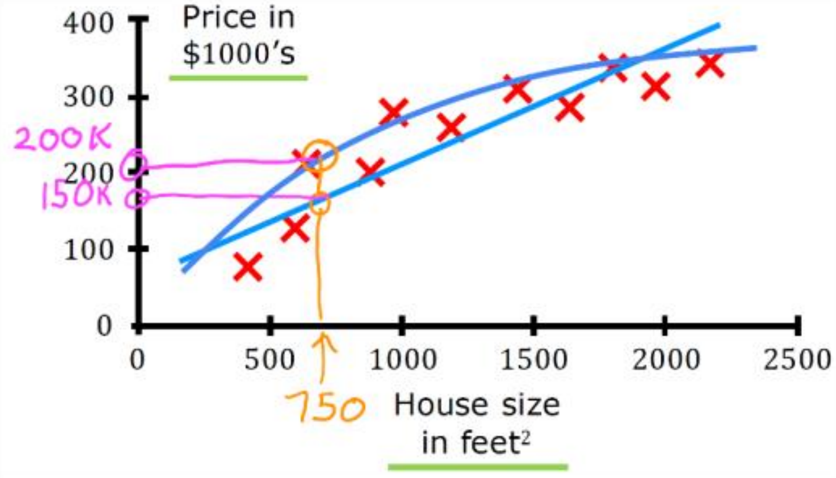
input → output label

learns from being given data labeled with the "right answers"

Regression

Predict a number

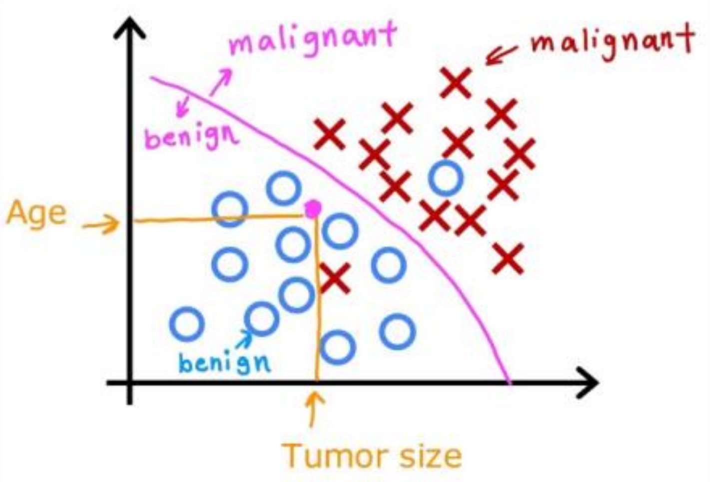
Infinitely many possible outcomes



Classification

Predict categories

Small number of possible outputs



RAPPEL

Unsupervised Learning

Data only comes with inputs x, but not output labels y
Algorithm has to find structure (= something interesting in unlabeled data)

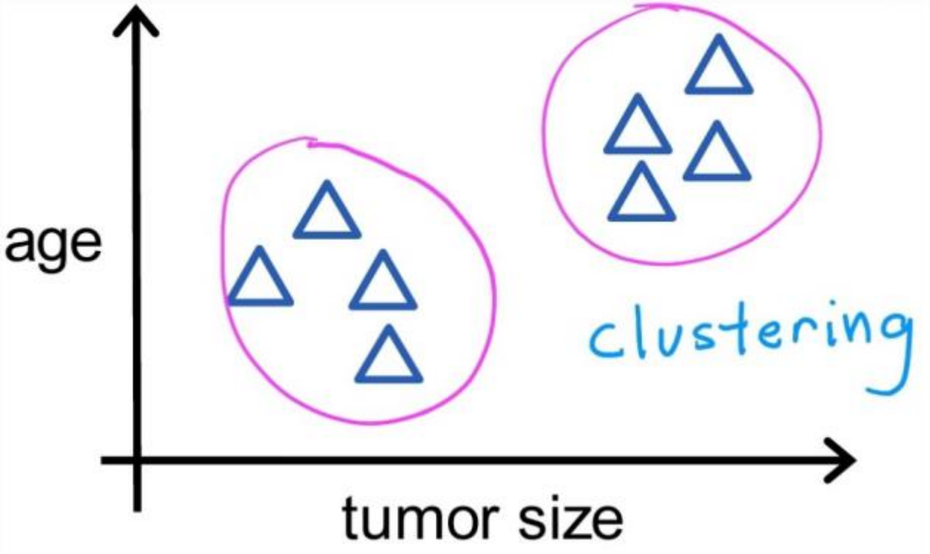
Clustering
Group similar data points together.

Anomaly detection
Find unusual data points.

Dimensionality reduction
Compress data using fewer numbers.

Clustering

Example : Google News



RAPPEL

Linear Regression Model

Terminology

x	y
$x^{(1)}$	$y^{(1)}$
$\vdots$	$\vdots$
$x^{(m)}$	$y^{(m)}$

(training set)

x = "input" variable / feature

y = "output" variable / "target" variable

m = number of variables in the training set

w = weight / b = bias

(x, y) = single training example

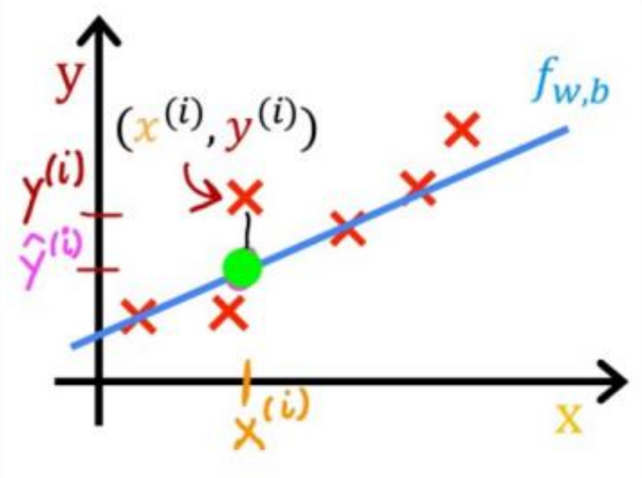
$(x^{(i)}, y^{(i)})$ = i^{th} training example

RAPPEL

Cost Function

$$\hat{y}^{(i)} = f_{w,b}(x^{(i)}) = w x^{(i)} + b$$

Find (w,b) such as $\hat{y}^{(i)} \approx y^{(i)} \ \forall (x,y)$



Squared error cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\big(\underbrace{f_{w,b}(x^{(i)})}_{\hat{y}^{(i)}} - y^{(i)}\big)^2$$

goal : to minimize $J(w,b)$

RAPPEL

Gradient Descent Algorithm

Repeat until convergence:

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

learning rate

Simultaneous update: ⚠

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$w = tmp\_w$$

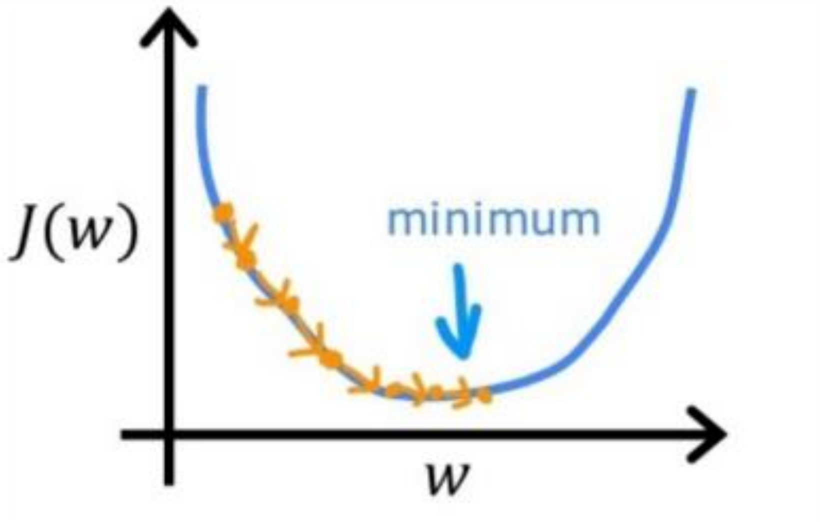
$$b = tmp\_b$$

Batch = each step of gradient descent uses all the training examples

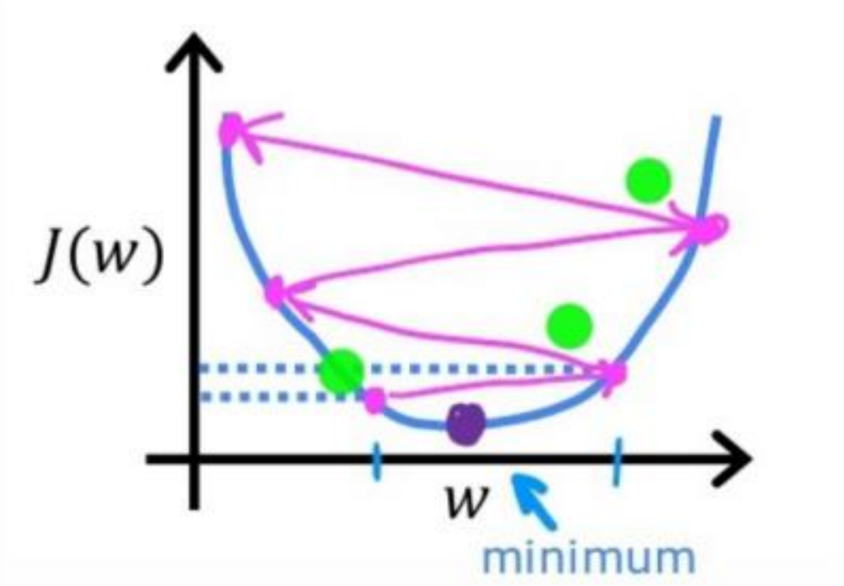
3 / TRAIN THE MODEL WITH GRADIENT DESCENT

RAPPEL

Learning Rate



If α is too small...
Gradient descent may be slow.



If α is too large...

Gradient descent may:
- Overshoot, never reach minimum
- Fail to converge, diverge

02 REGRESSION WITH MULTIPLE INPUT VARIABLES

INTRODUCTION TO MULTIPLE LINEAR REGRESSION

Multiple features
Vectorization
Lab : Python, NumPy and vectorization
Gradient descent for multiple linear regression
Lab : Multiple linear regression

GRADIENT DESCENT

Feature scaling
Checking gradient descent for convergence
Choosing the learning rate
Feature engineering
Polynomial regression
Lab : Feature scaling and learning rate
Lab : Feature engineering and polynomial regression
Lab : Linear regression with scikit-learn
Lab : Linear regression

01

INTRO TO MULTIPLE LINEAR REGRESSION

THÉO GACHET

Multiple Features

x_j = j^{th} feature

n = number of features

$\vec{x}^{(i)}$ = features of i^{th} training example

$x_j^{(i)}$ = value of feature j in i^{th} training example

Model

$$f_{\vec{w},b}(\vec{x}) = \sum_{i=1}^{n} w_i x_i + b = w_1 x_1 + \dots + w_n x_n$$

$$\left.\begin{aligned} \vec{x} &= [x_1 \quad x_2 \quad \dots \quad x_n] \\ \vec{w} &= [w_1 \quad w_2 \quad \dots \quad w_n] \end{aligned}\right\} \quad f_{\vec{w},b}(\vec{x}) = \vec{w} \overset{\text{dot product}}{\cdot} \vec{x} + b$$

Vectorization

Without vectorization

$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + \dots + w_n x_n$

```
f = 0
for i in range(n):
    f += w[i] * x[i]
f += b
```

☹

Vectorization

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

```
f = np.dot(w, x) + b
```

np.dot: multiplies in parallel

☺ shorter, faster

Gradient Descent

$\vec{w} = \vec{w} - \alpha \vec{d}$ where $\vec{d} = \left[\frac{\partial J_{w_1}}{\partial w_1} \; \frac{\partial J_{w_2}}{\partial w_2} \dots \frac{\partial J_{w_n}}{\partial w_n} \right]$

LAB-01

PYTHON, NUMPY AND VECTORIZATION

Gradient Descent for Multiple Linear Regression

Parameters

$\vec{w} = [w_1 \ \cdots \ w_n]$

b still a number

Model

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Cost

$J(\vec{w}, b)$

Gradient Descent

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

Gradient Descent

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

$j=1$ $$w_1 = w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_1^{(i)}$$

$\vdots$

$j=n$ $$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_n^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

Alternative to gradient descent : Normal Equation Method

Normal equation
- Only for linear regression
- Solve for w, b without iterations

Disadvantages
- Doesn't generalize to other learning algorithms.
- Slow when number of features is large (> 10,000)

What you need to know
- Normal equation method may be used in machine learning libraries that implement linear regression.
- Gradient descent is the recommended method for finding parameters w,b

LAB-02

MULTIPLE LINEAR REGRESSION

QUESTIONS ?

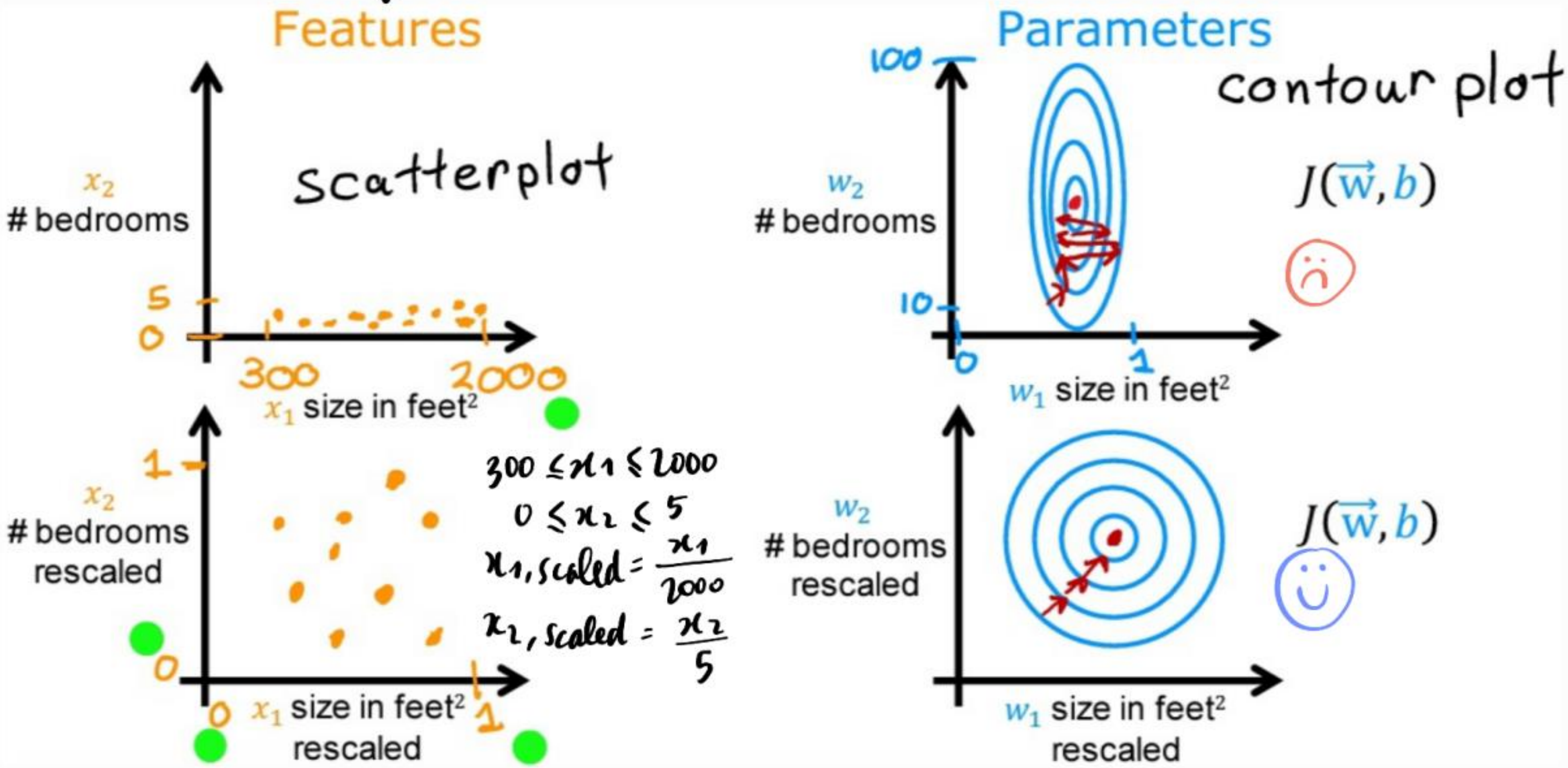
SUR UN CONCEPT ? UNE IDÉE ?
SUR UN DÉTAIL DU CODE ?
(ENVIE D'UNE PAUSE ?)

N'HÉSITEZ PAS !
IL N'Y A PAS DE QUESTION BÊTE, SI VOUS AVEZ UN DOUTE, D'AUTRES ONT SÛREMENT LE MÊME

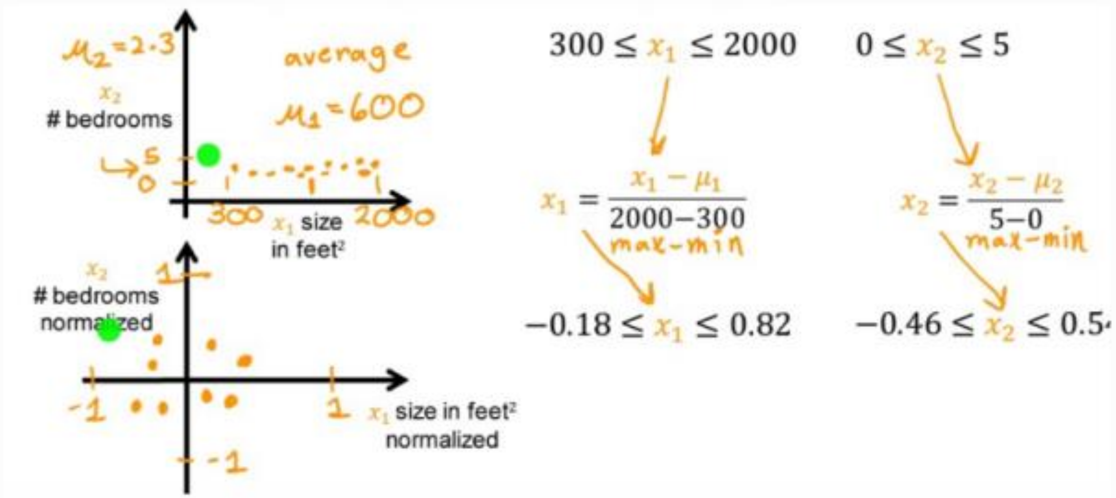
02

GRADIENT DESCENT IN PRACTICE

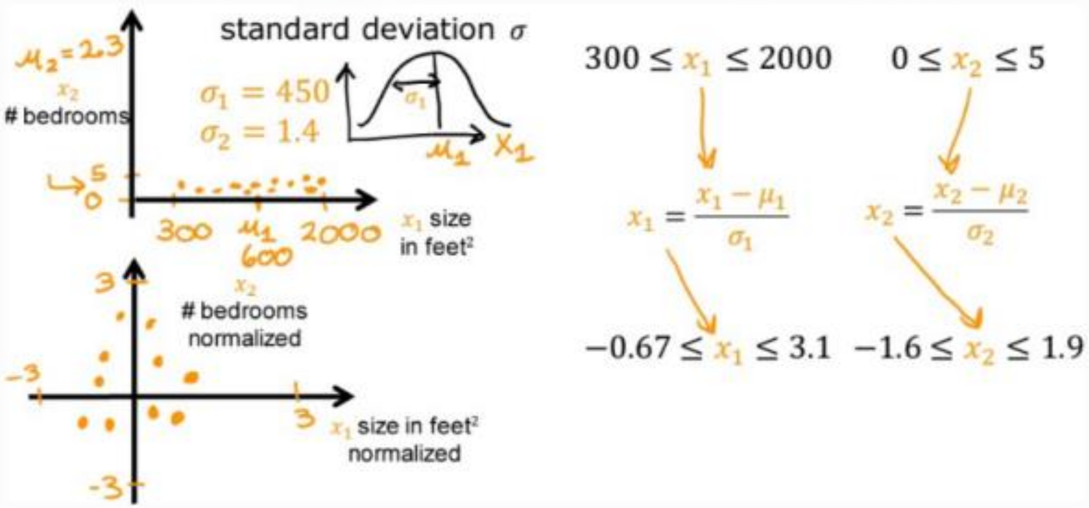
Feature Scaling



Mean Normalization



Z-score Normalization

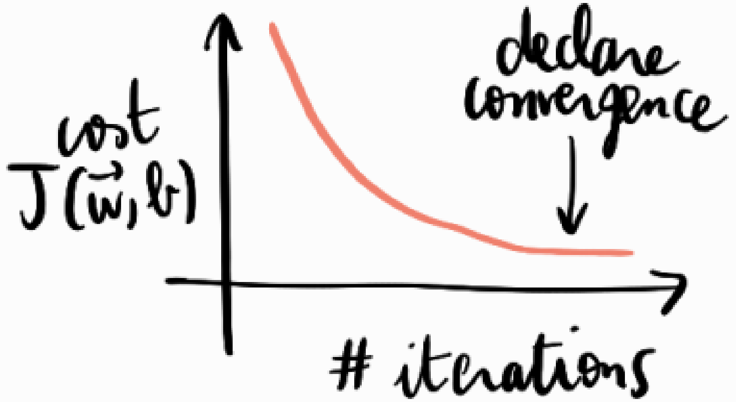


Feature Engineering Using intuition to design new features by using original features.

Example: if x_1 = frontage, x_2 = depth, then we can create $x_3 = x_1 \times x_2$ = area of the house, which is more relevant for price prediction $\Rightarrow$ $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3$

Gradient Descent Convergence

$J(\vec{w}, b)$ should decrease after each iteration



Automatic Convergence Test

let $\varepsilon = 10^{-3}$. If $J(\vec{w}, b)$ decreases by $\leq \varepsilon$ in 1 iteration, declare convergence
(found parameters $\vec{w}, b$ to get close to global minimum)

LAB-03

FEATURE SCALING AND LEARNING RATE

QUESTIONS ?

SUR UN CONCEPT ? UNE IDÉE ?
SUR UN DÉTAIL DU CODE ?
(ENVIE D'UNE PAUSE ?)

N'HÉSITEZ PAS !
IL N'Y A PAS DE QUESTION BÊTE, SI VOUS AVEZ UN DOUTE, D'AUTRES ONT SÛREMENT LE MÊME

LAB-04

FEATURE ENGINEERING AND POLYNOMIAL REGRESSION

THÉO GACHET

LAB-05

LINEAR REGRESSION WITH SCIKIT-LEARN

LAB-06

LINEAR REGRESSION

02 REGRESSION WITH MULTIPLE INPUT VARIABLES

INTRODUCTION TO MULTIPLE LINEAR REGRESSION

Multiple features
Vectorization
Lab : Python, NumPy and vectorization
Gradient descent for multiple linear regression
Lab : Multiple linear regression

GRADIENT DESCENT

Feature scaling
Checking gradient descent for convergence
Choosing the learning rate
Feature engineering
Polynomial regression
Lab : Feature scaling and learning rate
Lab : Feature engineering and polynomial regression
Lab : Linear regression with scikit-learn
Lab : Linear regression

UNE SÉRIE DE FORMATIONS

PARTIE 1 :
SUPERVISED MACHINE LEARNING : REGRESSION AND CLASSIFICATION

Introduction to Machine Learning
Regression with multiple input
Classification

PARTIE 2 :
ADVANCED LEARNING ALGORITHMS

Neural Networks
Neural Networks training
Advice for applying Machine Learning
Decision Trees

PARTIE 3 :
UNSUPERVISED LEARNING, RECOMMENDERS, REINFORCEMENT LEARNING

Unsupervised Learning
Recommander Systems
Reinforcement Learning

MINITEL

MACHINE LEARNING

02 REGRESSION WITH MULTIPLE INPUT VARIABLES

THÉO GACHET