Lecture 2
Linear Regression

LÊ ANH CƯỜNG

Recap from the previous lecture

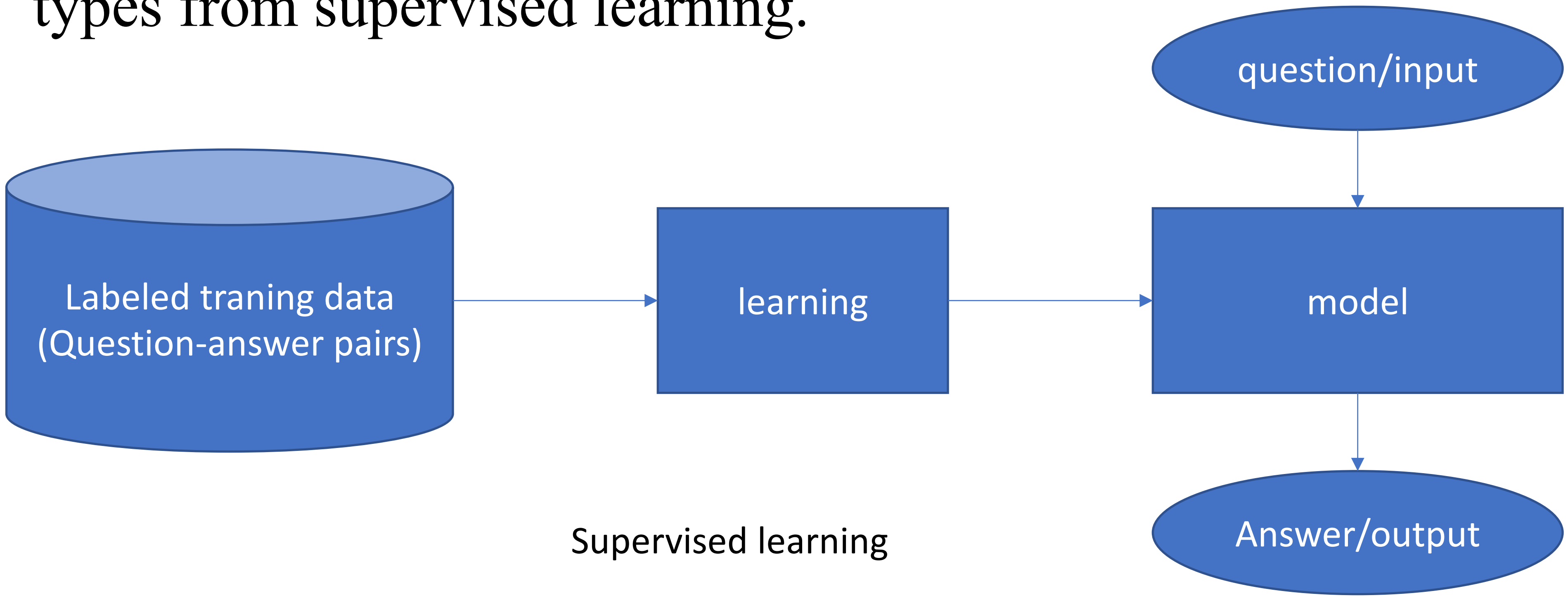
- Role of machine learning and artificial intelligence.
- A general model of AI systems.
- Concepts: knowledge, reasoning, inference, model, learning.
- Definition of machine learning.
- Understanding ML concepts through examples.

Outline

- Supervised learning for classification problems
- Linear Regression (LR) model
- Training LR by Gradient Descent Algorithm
- Implementation LR
- Exercises
- Summary

Supervised learning vs Unsupervised learning

- Supervised learning uses training data examples which are labeled as the objective of the task.
- Unsupervised learning doesn't use labled data, so it aims for different types from supervised learning.

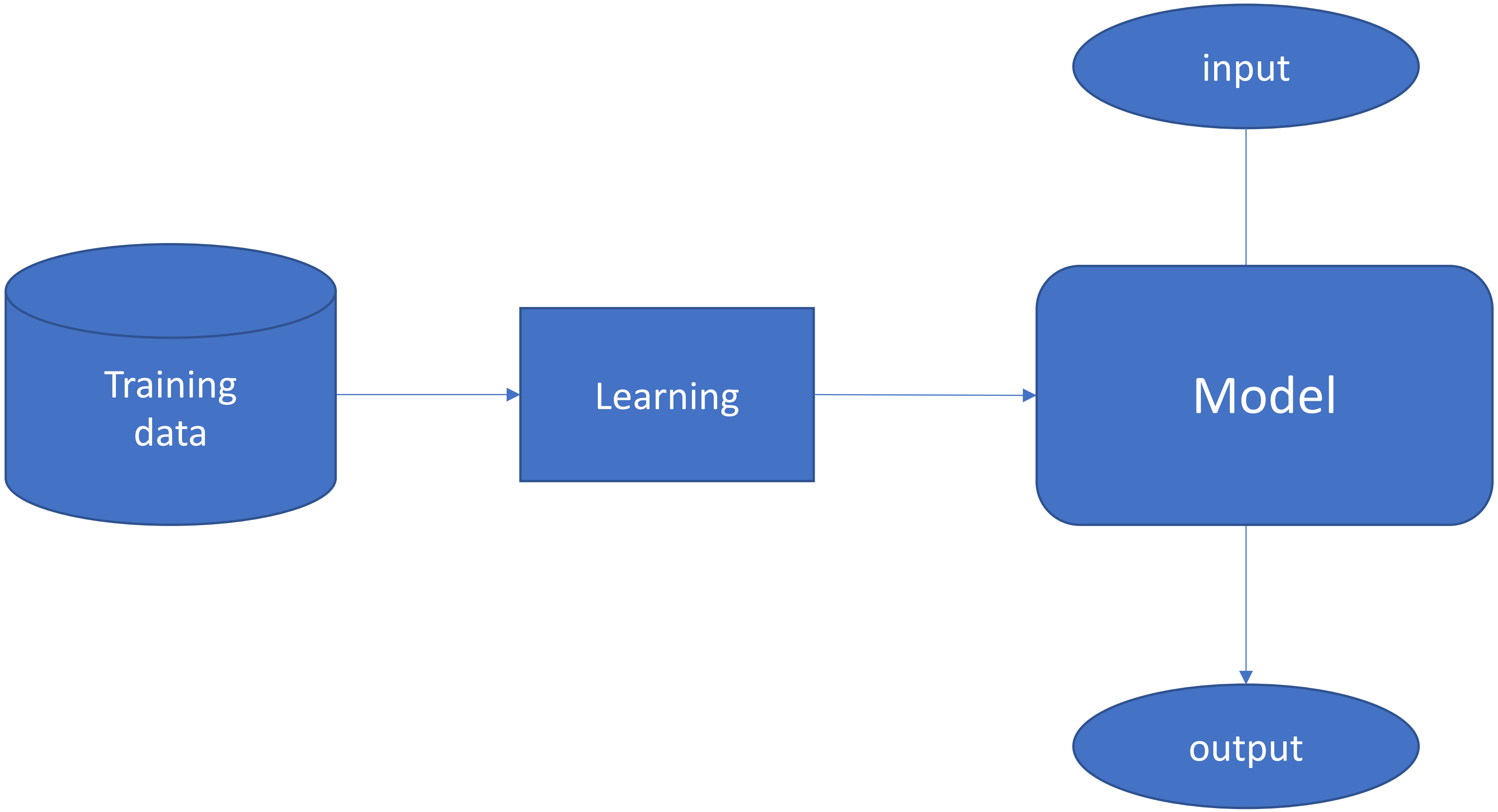


Supervised learning

Types of supervised learning problems?

- Classification
  - Single label
  - Multi-labels
- Generation
  - Labels
  - Real Values
- Sequence generation
  - Sequence of real values
  - Sequence of labels
- Structure Generation

Supervised Learning: a general model



General Mechanism of Supervised Learning

- Choose a model
  - The problem here is to determine parameters' values for the model
- Learning parameters of the model
- Inference/reasoning on the model

Repeat: Naïve Bayes Classification

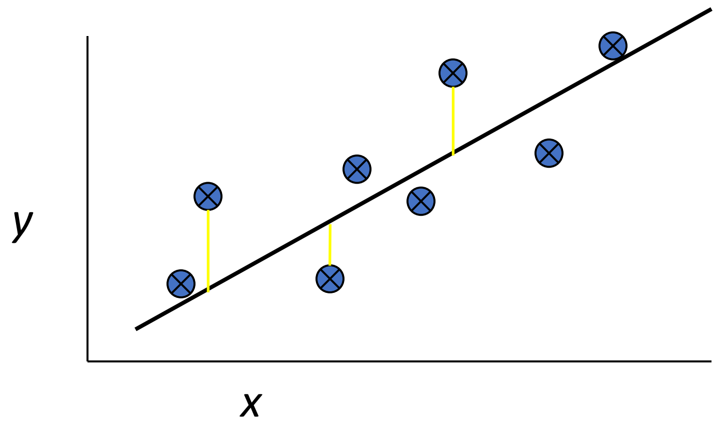
- Problem
- Model?
- Parameters of the model?

Regression vs Classification

- Distinguish between Regression and Classification
- Give examples?

Regression

- For classification the output(s) is nominal
- In regression the output is continuous
  - Function Approximation
- Many models could be used – Simplest is linear regression
  - Fit data with the best hyper-plane which "goes through" the points
  - For each point the differences between the predicted point and the actual observation is the *residue*



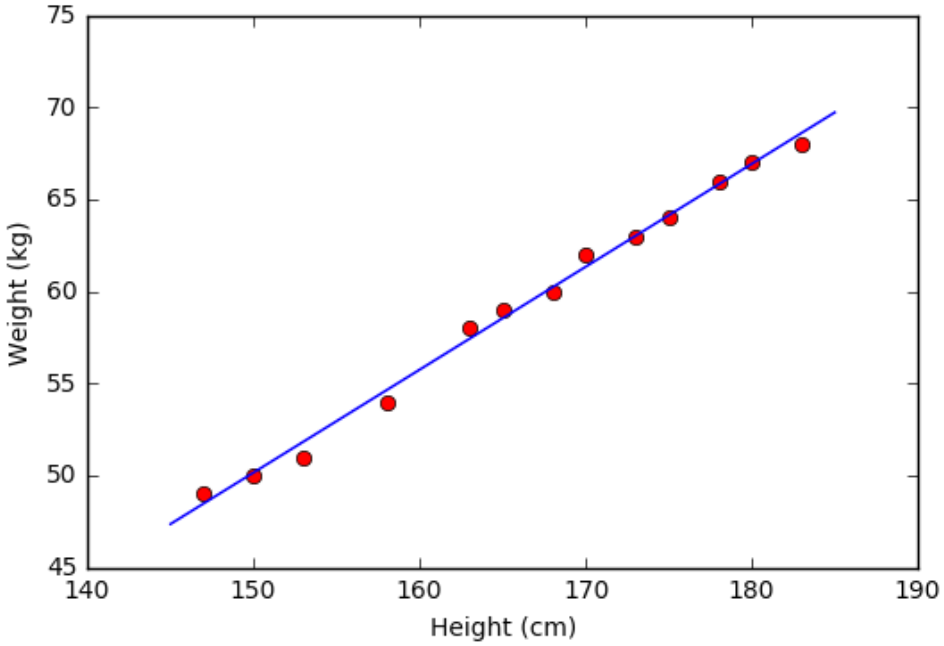
Regression (Hồi qui)

- From Height, predict Weight?

Height(cm)	Weight(kg)	Height(cm)	Weight(kg)
147	49	168	60
150	50	170	72
153	51	173	63
155	52	175	64
158	54	178	66
160	56	180	67
163	58	183	68
165	59		

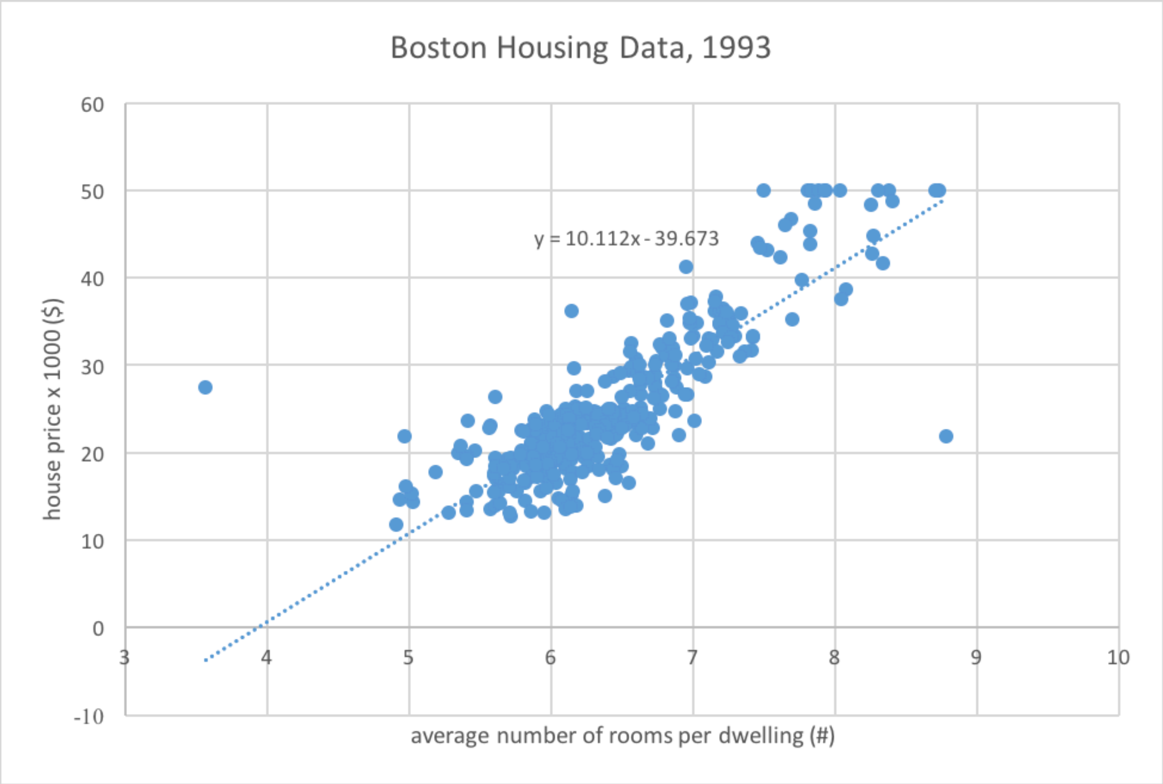
Linear Regression

- How is the relationship between Height and Weight?
- Suppose that Weight linearly depends on Height



Linear Regression

- Example of house pricing



Model Learning

- Choose a model form?
- Learning model's parameters
- Determine the loss function
- Learning: minimize the loss function then obtain parameters' values

Linear Regression: Formulation

Given a data set $\{y_i,\, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$ of n statistical units, a linear regression model assumes that the relationship between the dependent variable y and the p-vector of regressors $\mathbf{x}$ is linear. This relationship is modeled through a *disturbance term* or *error variable* ε — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus the model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where $^{\mathsf{T}}$ denotes the transpose, so that $\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}$ is the inner product between vectors $\mathbf{x}_i$ and $\boldsymbol{\beta}$.

Often these n equations are stacked together and written in matrix notation as

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

$$\varepsilon = y - X\beta$$

$$\varepsilon^2 = (y - X\beta)^2$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Simple Linear Regression

$y = f(\mathrm{x}) = w_0 + w_1 x$

$$\text{Loss} = L = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}((w_0 + w_1 x_i) - y_i)^2 =$$

Simple Linear Regression

$y = f(\mathrm{x}) = \ w_0 + \ w_1 x$

Loss $= L = \dfrac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 = \dfrac{1}{2N}\sum_{i=1}^{N}((w_0 + \ w_1 x_i) - y_i)^2 =$

Parameters: w_0, w_1

Goal: minimize Loss function

$$\frac{\partial L}{\partial w_0} = 0, \qquad \frac{\partial L}{\partial w_1} = 0$$

Linear Regression: Learning

Goal: minimize Loss function

$$\frac{\partial L}{\partial w_0} = 0, \qquad \frac{\partial L}{\partial w_1} = 0$$

$$L = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 =$$

$$= \frac{1}{2N}\sum_{i=1}^{N}((w_0 + w_1 x_i) - y_i)^2$$

$$\frac{\partial L}{\partial w_0} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)$$

$$\frac{\partial L}{\partial w_1} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)\, x_i$$

Muti-variables Linear regression

Simple Linear Regression

$(X_{\mathrm{i}}, \mathrm{y}_{\mathrm{i}})$

$X_{\mathrm{i}} = (x_0, \ldots, x_{\mathrm{k}})$

$y_{\mathrm{i}} = f(\mathrm{X}_{\mathrm{i}}) = w_0 x_{\mathrm{i}0} + w_1 x_{\mathrm{i}1} + \ldots + w_{\mathrm{k}} x_{\mathrm{i}k}$

$$\text{Loss} = L = \frac{1}{2N}\sum_{i=1}^{N}(f(\mathrm{X}) - y_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}((w_0 + w_1 x) - y_i)^2 =$$

$$\frac{\partial L}{\partial w_{\mathrm{t}}} = \frac{1}{N}\sum_{i=1}^{N}(f(\mathrm{X}_i) - y_i)\, x_{i\mathrm{t}}$$

Simple Linear Regression

(X_i, y_i)

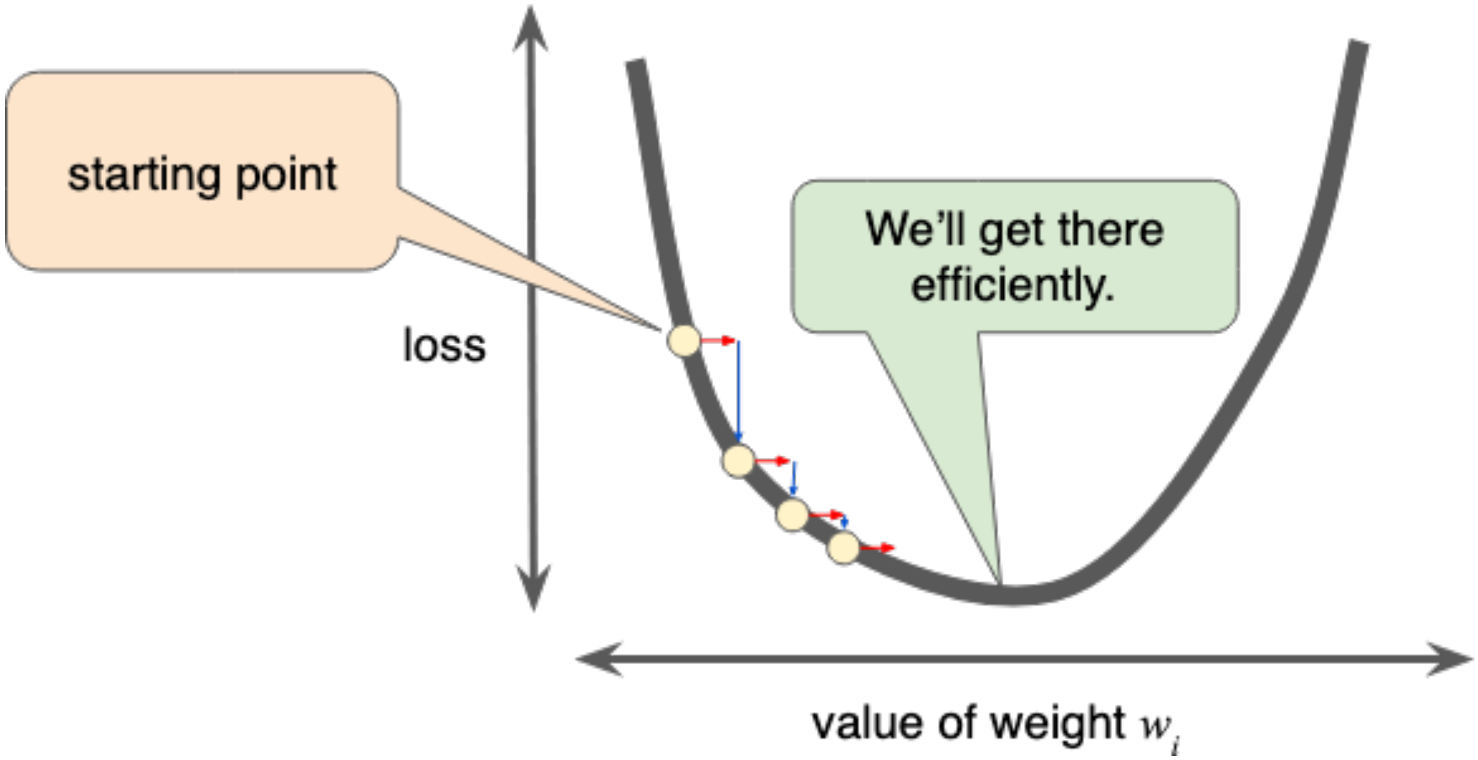
$X_i = (x_0, \ldots, x_k)$

$y_i = f(X_i) = w_0 x_{i0} + w_1 x_{i1} + \ldots + w_k x_{ik}$

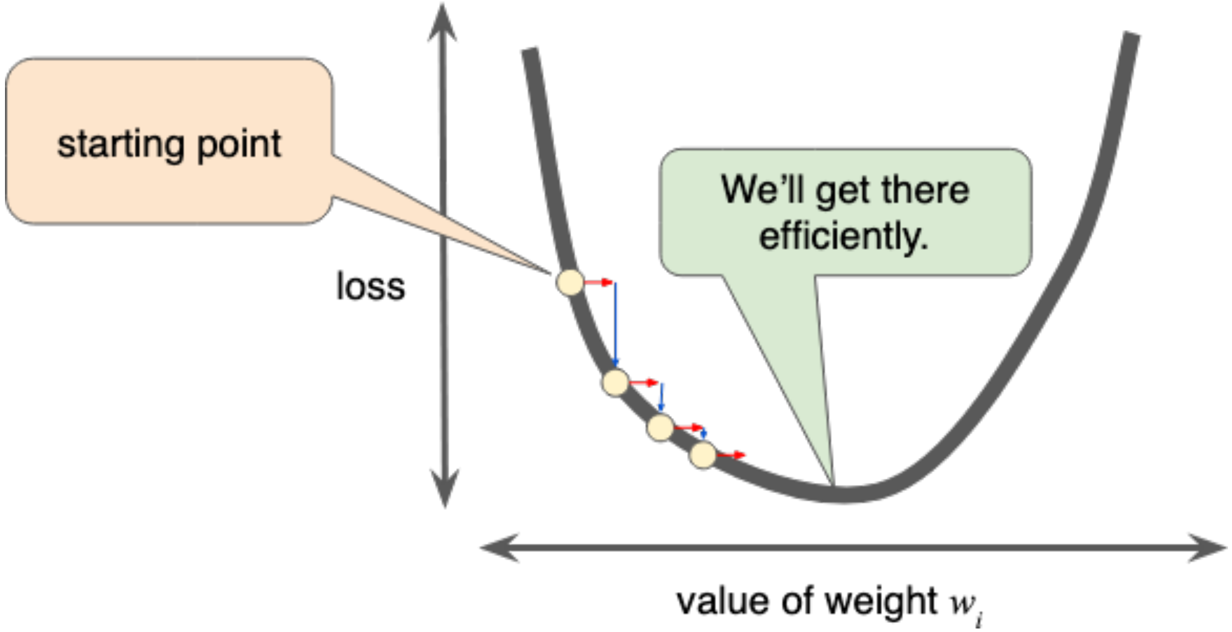
$$\text{Loss} = L = \frac{1}{2N}\sum_{i=1}^{N}(f(X) - y_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}((w_0 + w_1 x) - y_i)^2 =$$

$$\frac{\partial L}{\partial w_t} = \frac{1}{N}\sum_{i=1}^{N}(f(X_i) - y_i)\, x_{it}$$

Linear Regression: Learning by Gradient Descent



Linear Regression: Learning by Gradient Descent

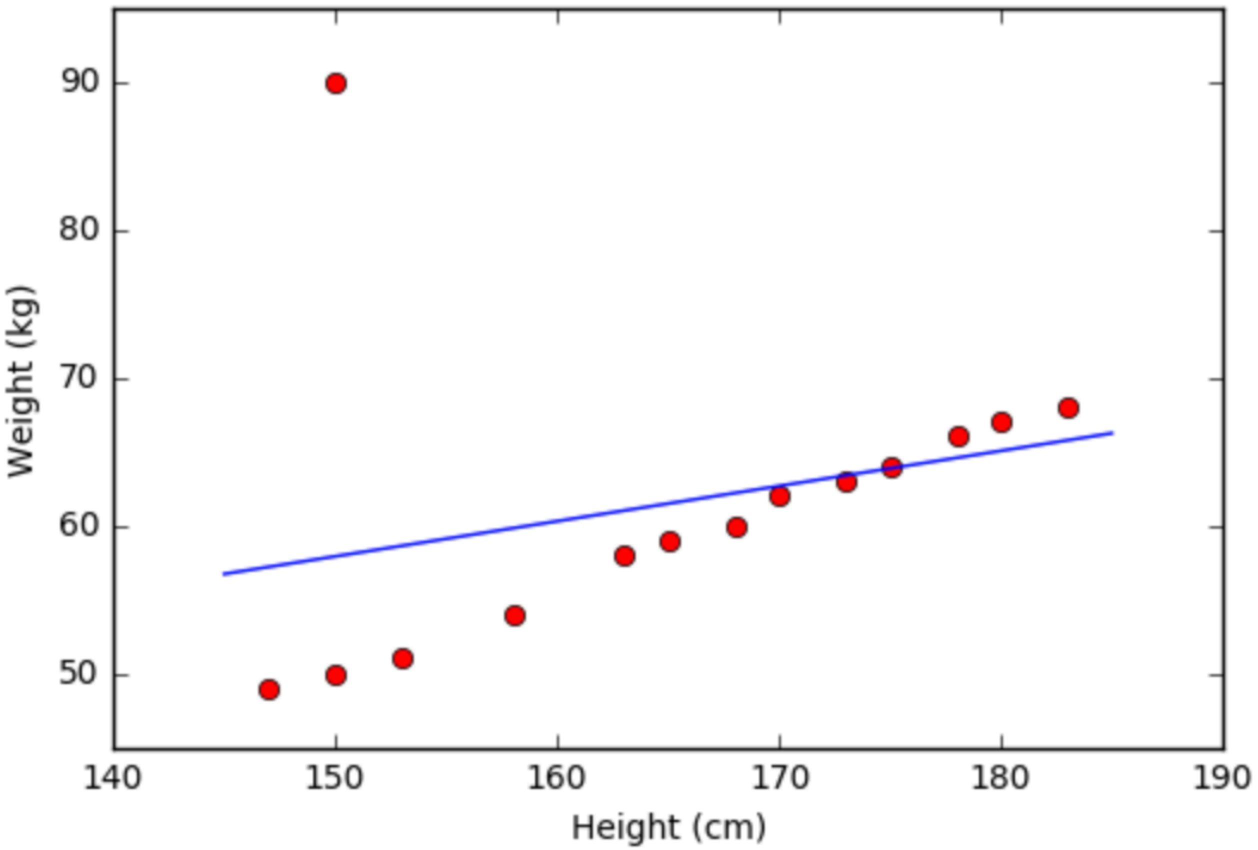


Update parameters:

Learning rate

$$w_i = w_i - \mu \frac{\partial L}{\partial w_i}$$

Linear Regression is sensitive with outliers



Summary

- Supervised learning vs unsupervised learning.
- Classification vs regression
- Linear regression form.
- Learning Linear Regression model by Gradient Descent algorithm?

Exercise

- Implement Linear Regression for the problem of house pricing with multiple variables:
    - Firstly derive steps and mathematical formulas in the algorithm.
    - Secondly, implementation using python.

Answer the questions:

1. What is the objective of a supervised learning model? General model?
2. What is linear regression model?
3. Formulating the linear regression model with single variable?
   - Function y = f(x)
   - Loss function
4. Learning parameters by Gradient Descent
   - Update parameters?
   - Learning rate?
5. Implementation
6. What are limitations of linear regression

```python
# hàm y = f(x)
# w: tham số
def f(x,w):
    return w[0]+w[1]*x

# tính hàm loss
def loss(x,y,w):
    d = 0
    for i in range(len(x)):
        d += (y[i] - (w[0] + w[1]*x[i]))**2
    return d/(2*len(x))

# tính đạo hàm tại điểm w[0] và w[1]
def derivative(x,y,w):
    d0=0
    d1=0
    for i in range(len(x)):
        d1 += x[i]*(f(x[i],w)-y[i])
        d0 += f(x[i],w)-y[i]
    return d0/len(x),d1/len(x)
```

```python
# training
epoch = 10
learning_rate = 0.01
w = [1,1] # y = x + 1
los_old = 0
for i in range(epoch):
    # hiển thị đồ thị
    plt.plot(x,y,'ro')
    plt.xlabel('X')
    plt.ylabel('Y')
    x0 = np.linspace(start=1, stop=10, num=50)
    y0 = w[0]+w[1]*x0
    plt.plot(x0,y0)
    plt.show()

    # cập nhật tham số
    los = loss(x,y,w)
    print('epoch_',i,':')
    print(los,' : ',los_old)
    if los>(los_old-0.0001) and i>0:
        break
    los_old = los

    # cập nhật
    a,b = derivative(x,y,w)# cho w0 và w1
    w[0] = w[0]-a*learning_rate
    w[1] = w[1]-b*learning_rate
```