AI VIETNAM
Free AI Course 2020

Computational Graph and Linear Regression (Draft)

Quang-Vinh Dinh
Ph.D. in Computer Science

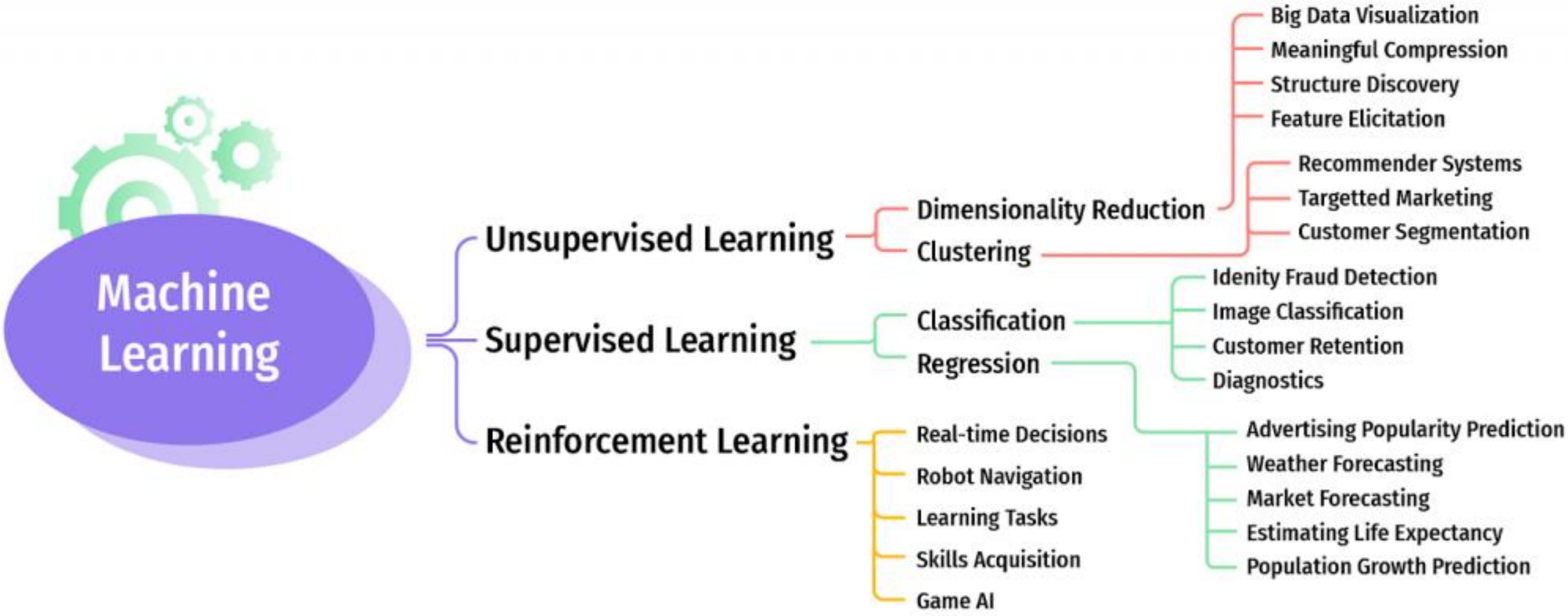
Year 2020

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formula

Machine Learning

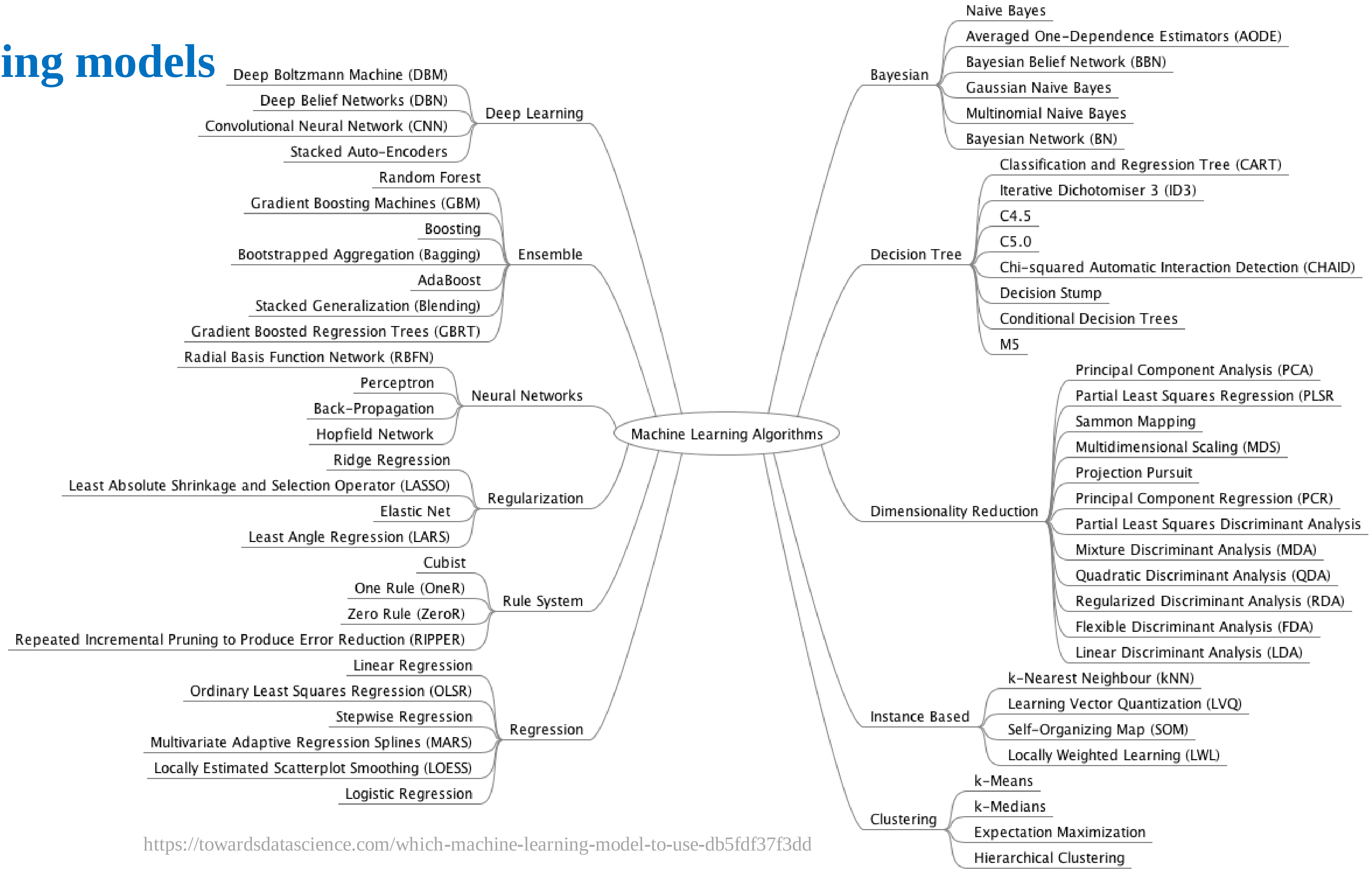
❖ Overview



https://idapgroup.com/blog/types-of-machine-learning-out-there/

Machine Learning

Machine learning models



- Machine Learning Algorithms
  - Deep Learning
    - Deep Boltzmann Machine (DBM)
    - Deep Belief Networks (DBN)
    - Convolutional Neural Network (CNN)
    - Stacked Auto-Encoders
  - Ensemble
    - Random Forest
    - Gradient Boosting Machines (GBM)
    - Boosting
    - Bootstrapped Aggregation (Bagging)
    - AdaBoost
    - Stacked Generalization (Blending)
    - Gradient Boosted Regression Trees (GBRT)
  - Neural Networks
    - Radial Basis Function Network (RBFN)
    - Perceptron
    - Back-Propagation
    - Hopfield Network
  - Regularization
    - Ridge Regression
    - Least Absolute Shrinkage and Selection Operator (LASSO)
    - Elastic Net
    - Least Angle Regression (LARS)
  - Rule System
    - Cubist
    - One Rule (OneR)
    - Zero Rule (ZeroR)
    - Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
  - Regression
    - Linear Regression
    - Ordinary Least Squares Regression (OLSR)
    - Stepwise Regression
    - Multivariate Adaptive Regression Splines (MARS)
    - Locally Estimated Scatterplot Smoothing (LOESS)
    - Logistic Regression
  - Bayesian
    - Naive Bayes
    - Averaged One-Dependence Estimators (AODE)
    - Bayesian Belief Network (BBN)
    - Gaussian Naive Bayes
    - Multinomial Naive Bayes
    - Bayesian Network (BN)
  - Decision Tree
    - Classification and Regression Tree (CART)
    - Iterative Dichotomiser 3 (ID3)
    - C4.5
    - C5.0
    - Chi-squared Automatic Interaction Detection (CHAID)
    - Decision Stump
    - Conditional Decision Trees
    - M5
  - Dimensionality Reduction
    - Principal Component Analysis (PCA)
    - Partial Least Squares Regression (PLSR
    - Sammon Mapping
    - Multidimensional Scaling (MDS)
    - Projection Pursuit
    - Principal Component Regression (PCR)
    - Partial Least Squares Discriminant Analysis
    - Mixture Discriminant Analysis (MDA)
    - Quadratic Discriminant Analysis (QDA)
    - Regularized Discriminant Analysis (RDA)
    - Flexible Discriminant Analysis (FDA)
    - Linear Discriminant Analysis (LDA)
  - Instance Based
    - k-Nearest Neighbour (kNN)
    - Learning Vector Quantization (LVQ)
    - Self-Organizing Map (SOM)
    - Locally Weighted Learning (LWL)
  - Clustering
    - k-Means
    - k-Medians
    - Expectation Maximization
    - Hierarchical Clustering

https://towardsdatascience.com/which-machine-learning-model-to-use-db5fdf37f3dd

Machine Learning

❖ Supervised learning

Input and output data is provided

🟩 Training data

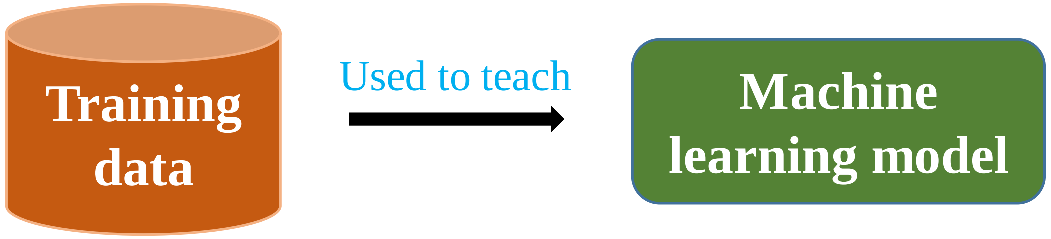
🟥 Cats

🟪 Dogs

From Cat-Dog dataset

Machine Learning

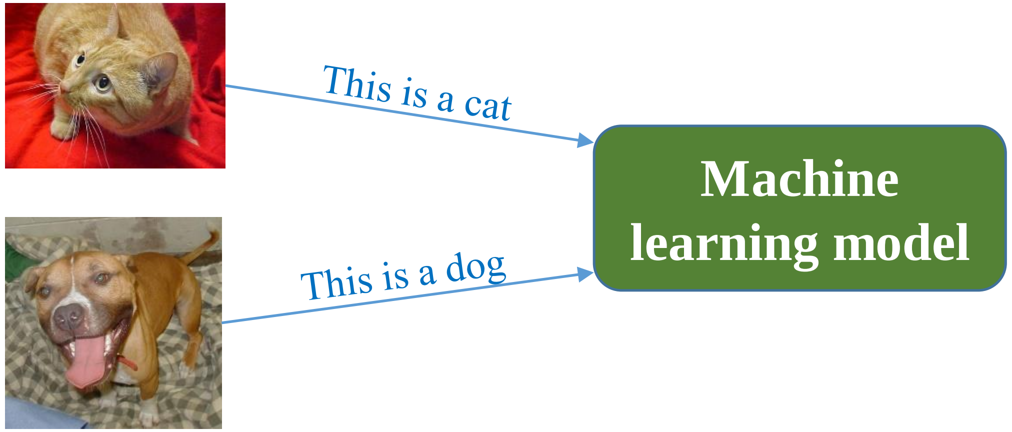
❖ Supervised learning



Training data

Used to teach

Machine learning model



This is a cat

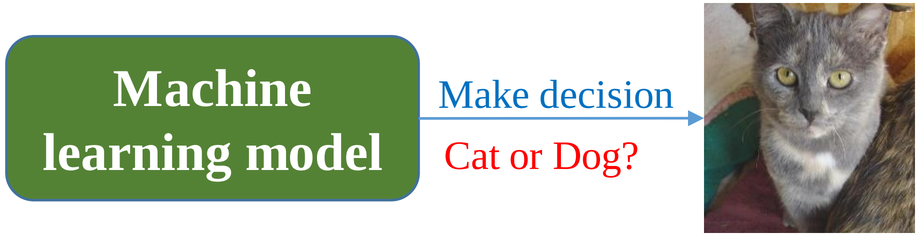
This is a dog

Machine learning model

Training phase

From Cat-Dog dataset

Testing data (≠ training data)



Machine learning model

Make decision

Cat or Dog?

Testing phase

Machine Learning

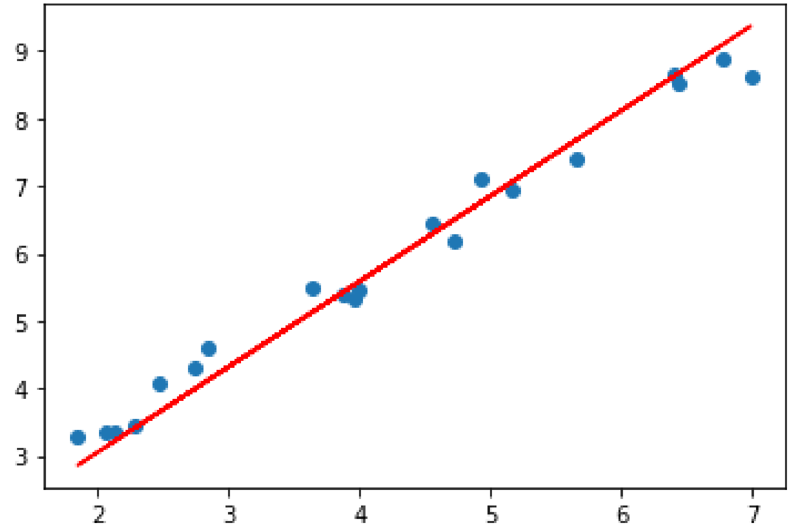
❖ Supervised learning

❖Regression (prediction)

Linear regression models ← Linear equations

Linear equation = $w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$

where $\boldsymbol{w}$ is a weight vector
and $\boldsymbol{x}$ is feature vector

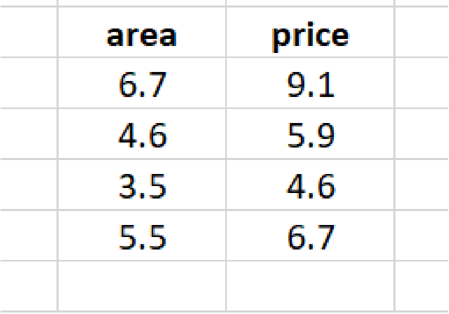


Machine Learning

❖ Supervised learning

❖ Linear Regression: Data processing

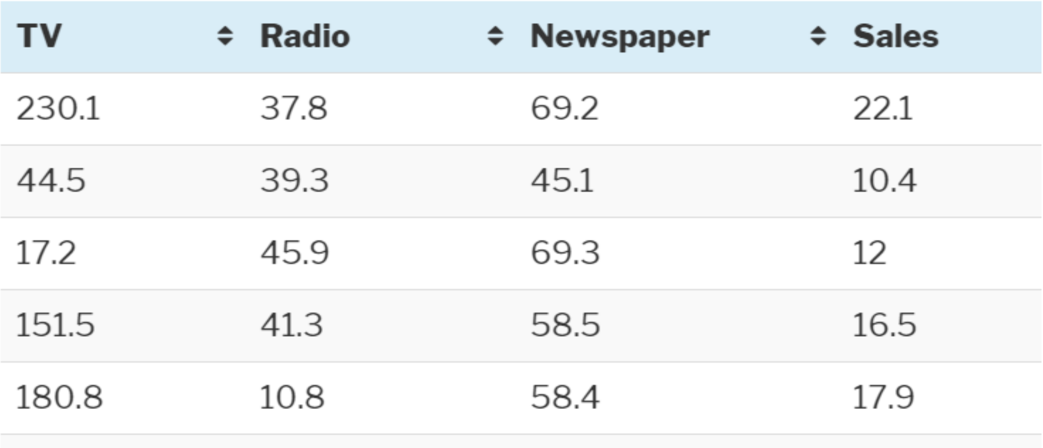
Feature Label



area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

House price data

Features Label



TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Advertising data

Model: $\mathrm{y} = w_1x_1 + b$

$\mathrm{price} = \mathrm{a} * area + b$

Model: $\mathrm{y} = w_1x_1 + w_2x_2 + w_3x_3 + b$

$\mathrm{Sale} = w_1 * TV + w_2 * Radio + w_3 * Newspaper + b$

Machine Learning

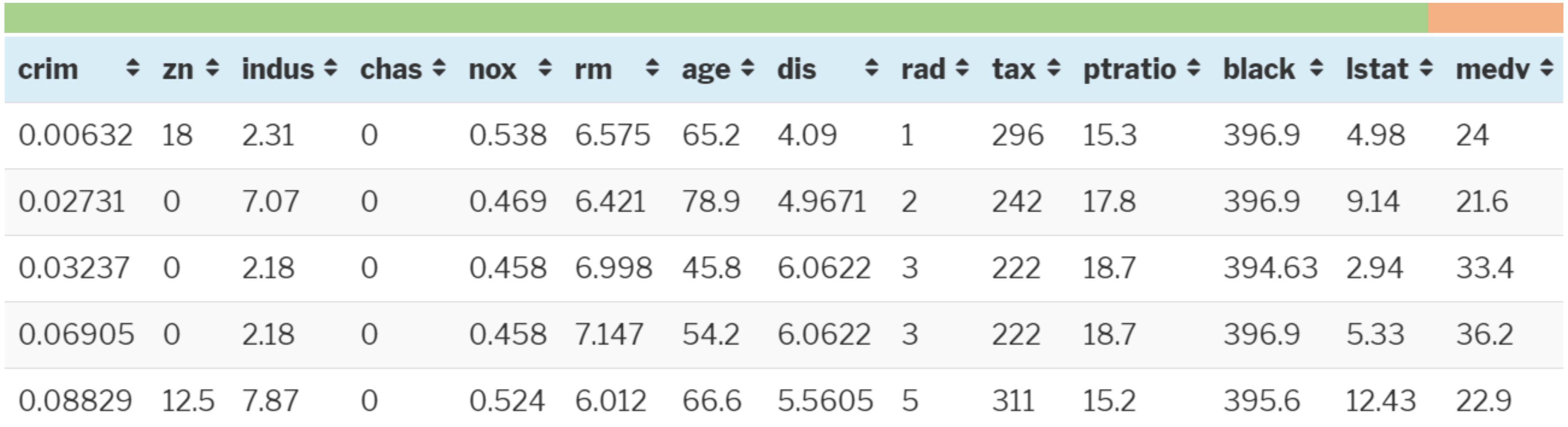
❖ Supervised learning

❖ Linear Regression: Data processing

Features

Label

Boston House Price Data



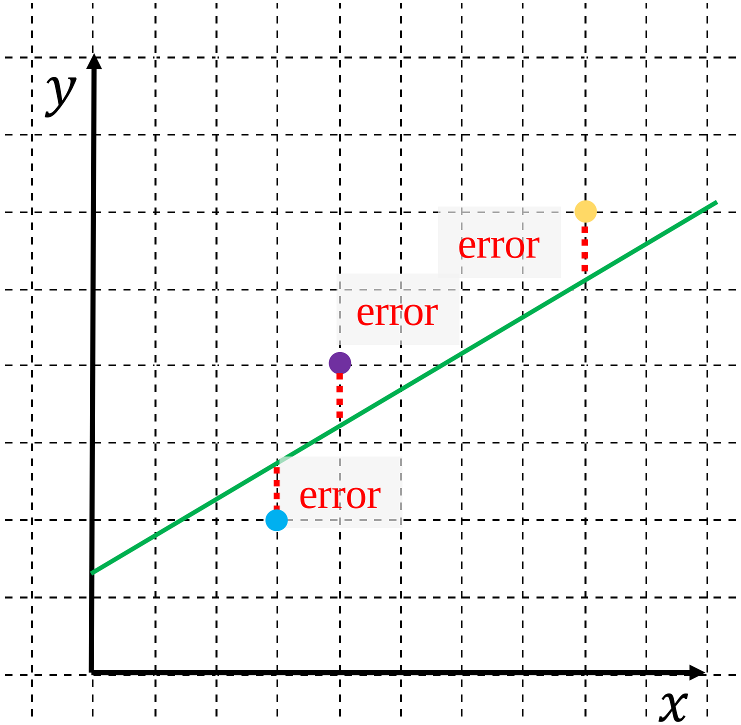
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9

$$\text{medv} = w_1 * x_1 + \cdots + w_{13} * x_{13} + b$$

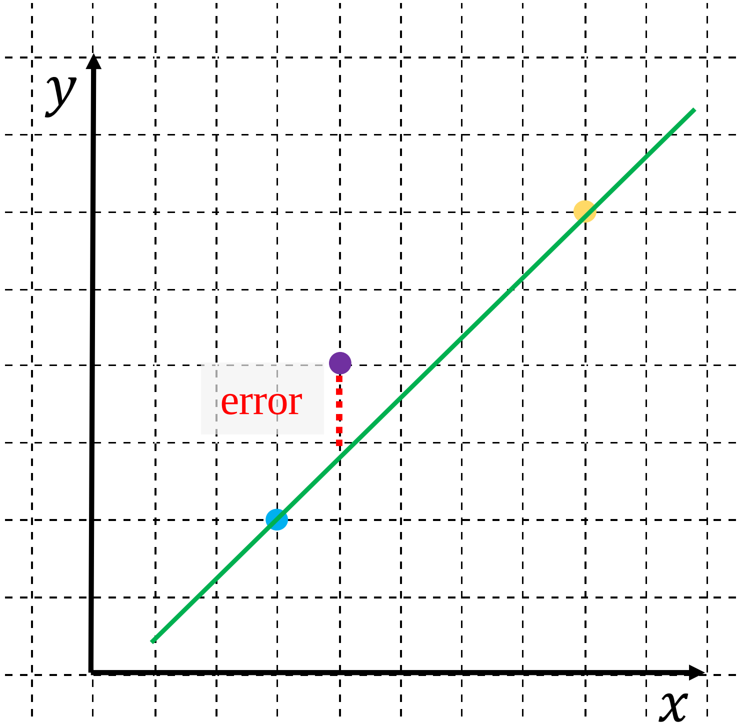
Machine Learning

- **Linear regression**
  - **Aim to fit data**

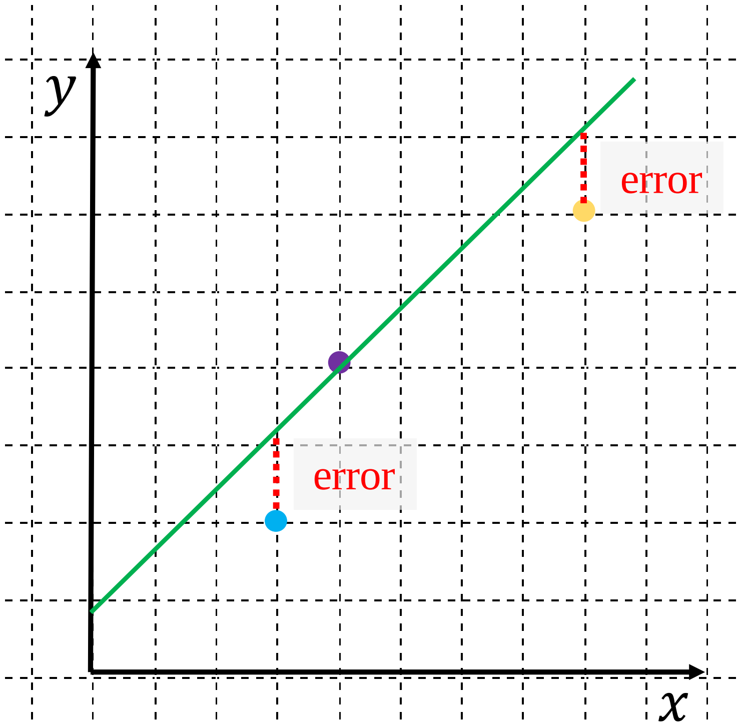
● ● ● Training data



$y = a_1 x + b_1$



$y = a_2 x + b_2$



$y = a_3 x + b_3$

Find w and b that have the smallest error.

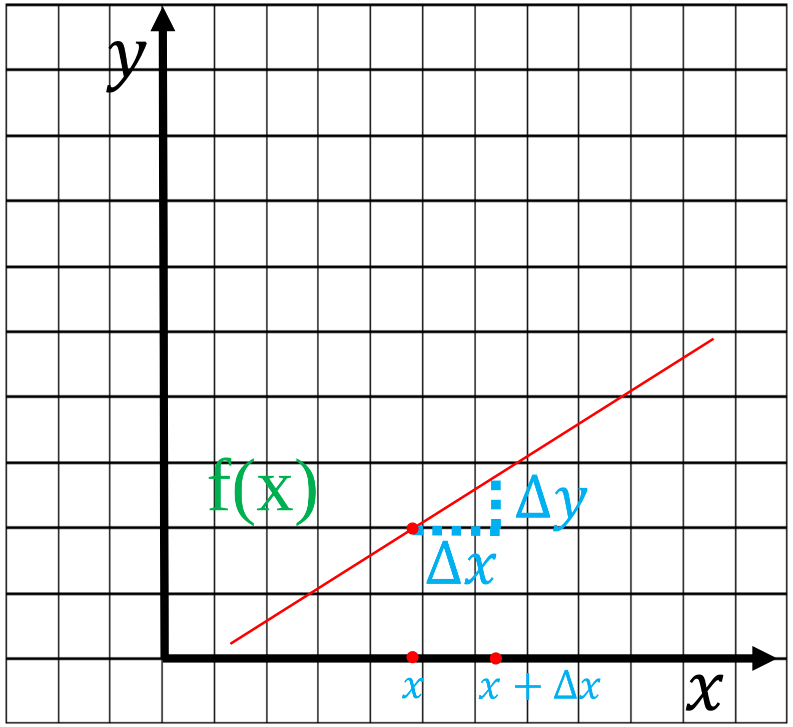
How?

Outline

- ➢ Machine Learning
- ➢ Derivative/Gradient
- ➢ Linear Regression
- ➢ Computational Graph
- ➢ Generalized formula

Derivative/Gradient

❖ A cue to optimize a function



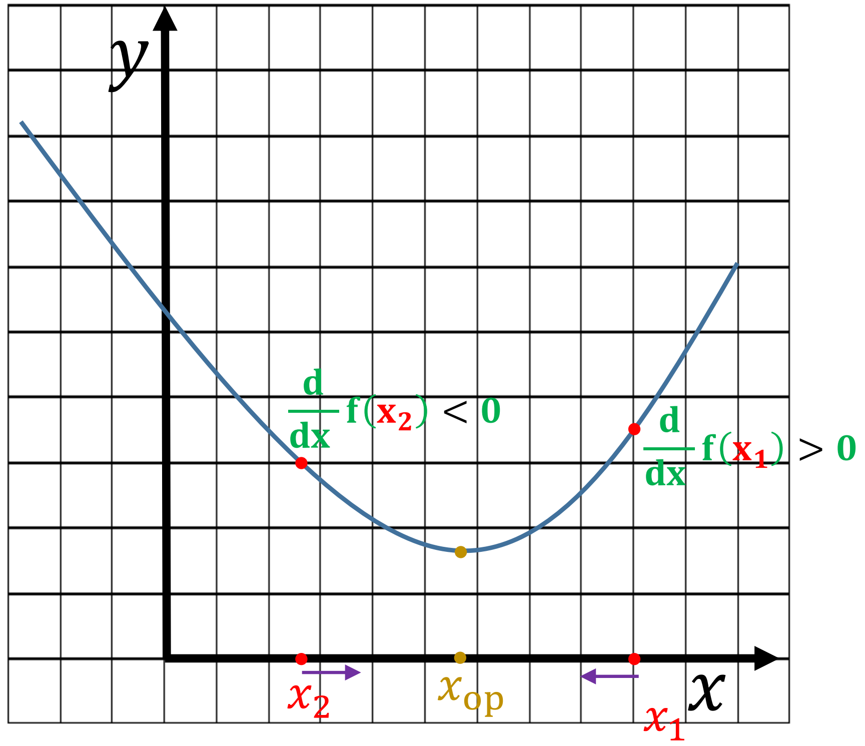
$$\text{Đạo hàm} = \frac{\textit{Thay đổi theo } y}{\textit{Thay đổi theo } x} = \frac{\Delta y}{\Delta x}$$

$$\frac{d}{dx} f(x) = \underbrace{\lim_{\Delta x \to 0}} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

Δx cần tiến về 0 để đường tiếp tuyến tiến về hàm f(x) trong vùng lân cận tại x

Derivative/Gradient

❖ A cue to optimize a function



Quan sát: x_{op} ở vị trí ngược hướng đạo hàm tại $\mathbf{x_1}$ và $\mathbf{x_2}$

Cách xử lý việc di chuyển ngược hướng đạo hàm cho $\mathbf{x_1}$ và $\mathbf{x_2}$ (để tìm x_{op}) khác nhau hình thành các thuật toán tối ưu hóa khác nhau

Cách cập nhật giá trị x đơn giản

$$\mathbf{x} = \mathbf{x} - \eta \frac{\mathbf{d}}{\mathbf{dx}}\mathbf{f}(\mathbf{x})$$

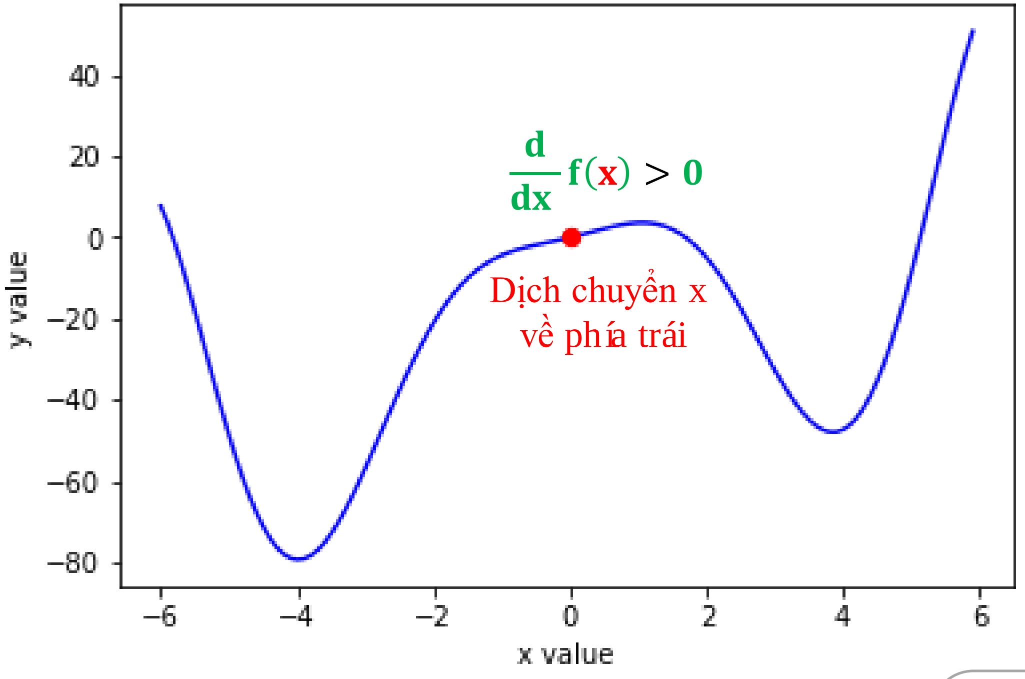
Đạo hàm tại x

Trọng số

Derivative/Gradient

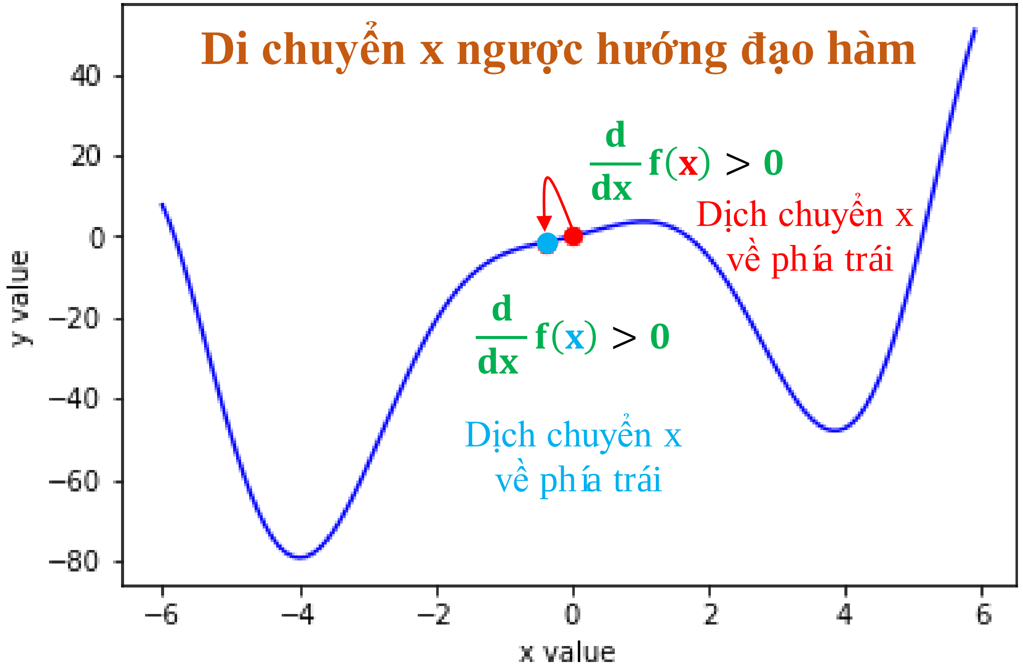
❖ A cue to optimize a function

Khởi tạo giá trị x



$$\frac{d}{dx}f(x) > 0$$

Dịch chuyển x
về phía trái



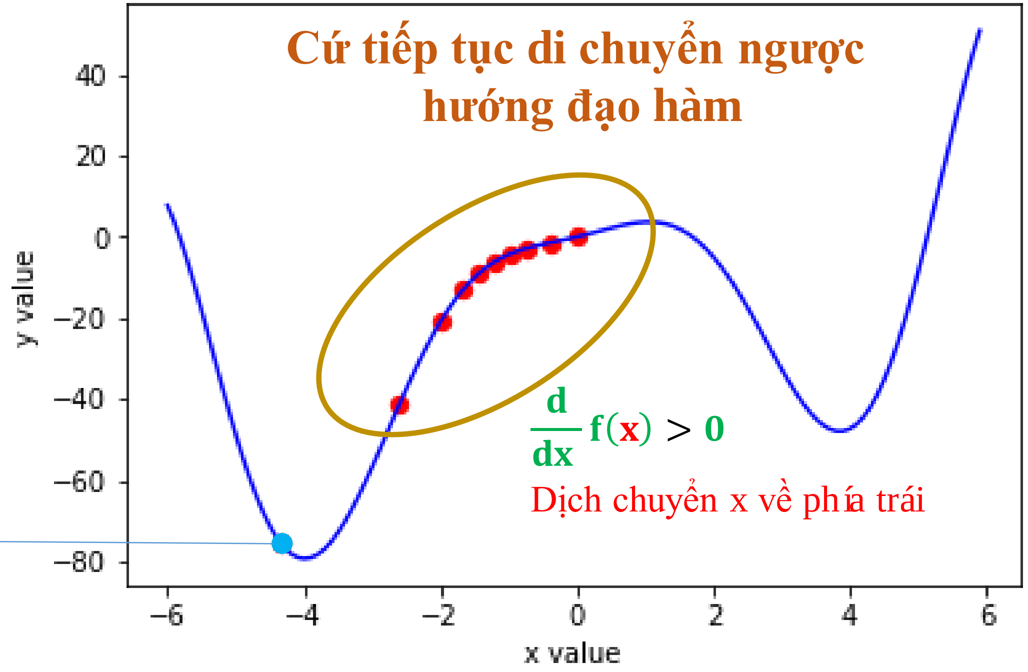
Di chuyển x ngược hướng đạo hàm

$$\frac{d}{dx}f(x) > 0$$

Dịch chuyển x
về phía trái

$$\frac{d}{dx}f(x) > 0$$

Dịch chuyển x
về phía trái



Cứ tiếp tục di chuyển ngược
hướng đạo hàm

$$\frac{d}{dx}f(x) > 0$$

Dịch chuyển x về phía trái

$$\frac{d}{dx}f(x) < 0$$

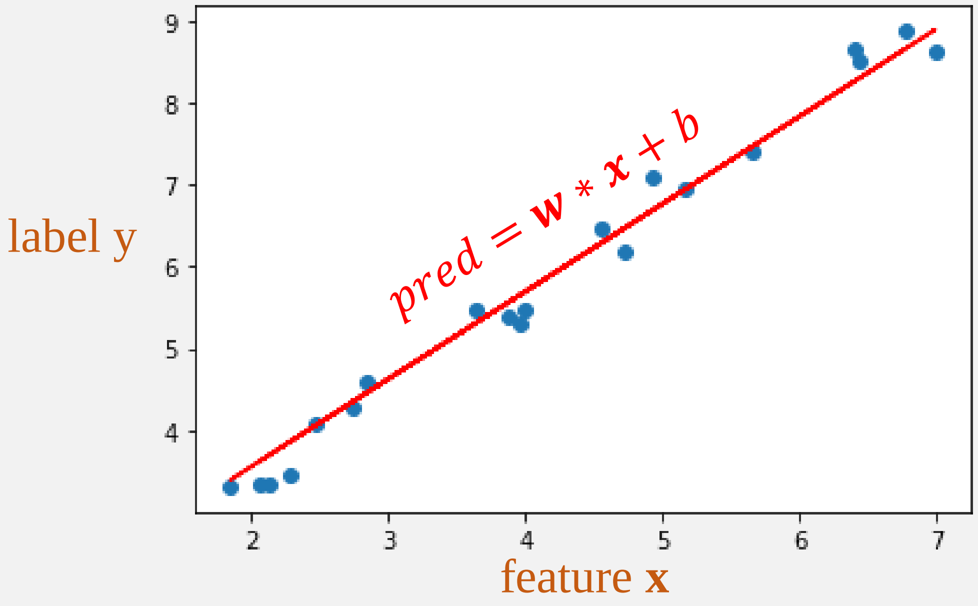
Dịch chuyển x
về phía phải

Outline

- ➢ Machine Learning
- ➢ Derivative/Gradient
- ➢ Linear Regression
- ➢ Computational Graph
- ➢ Generalized formula

Linear Regression

Model the relationship between feature x and label y



Using a linear equation to fit data

Samples (x, y) are given in advance

Linear equation

$$o = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

where o is a predicted value,
$w_1, w_2, \ldots, w_n$ and b are parameters
and $\boldsymbol{x} = [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$ is feature vector.

Error (loss) computation

Idea: compare predicted values o and label values y

Squared loss

$$\mathrm{L}(\mathbf{w}, \mathrm{b}) = (o - y)^2$$

How to find optimal w and b?

Linear Regression

Linear equation

$$o = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

where o is a predicted value,
$w_1, w_2, \ldots, w_n$ and b are parameters
and $\boldsymbol{x} = [x_1 \ \ x_2 \ \ \ldots \ \ x_n]^T$ is feature vector.

Error (loss) computation

Idea: compare predicted values o and label values y

Squared loss

$$\mathrm{L}(\mathbf{w}, \mathrm{b}) = (o - y)^2$$

How to find optimal **w** and b?

Use gradient descent to minimize the loss function

Tính đạo hàm

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_j} = 2x_j(o - y)$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_j} = 2(o - y)$$

Cập nhật tham số

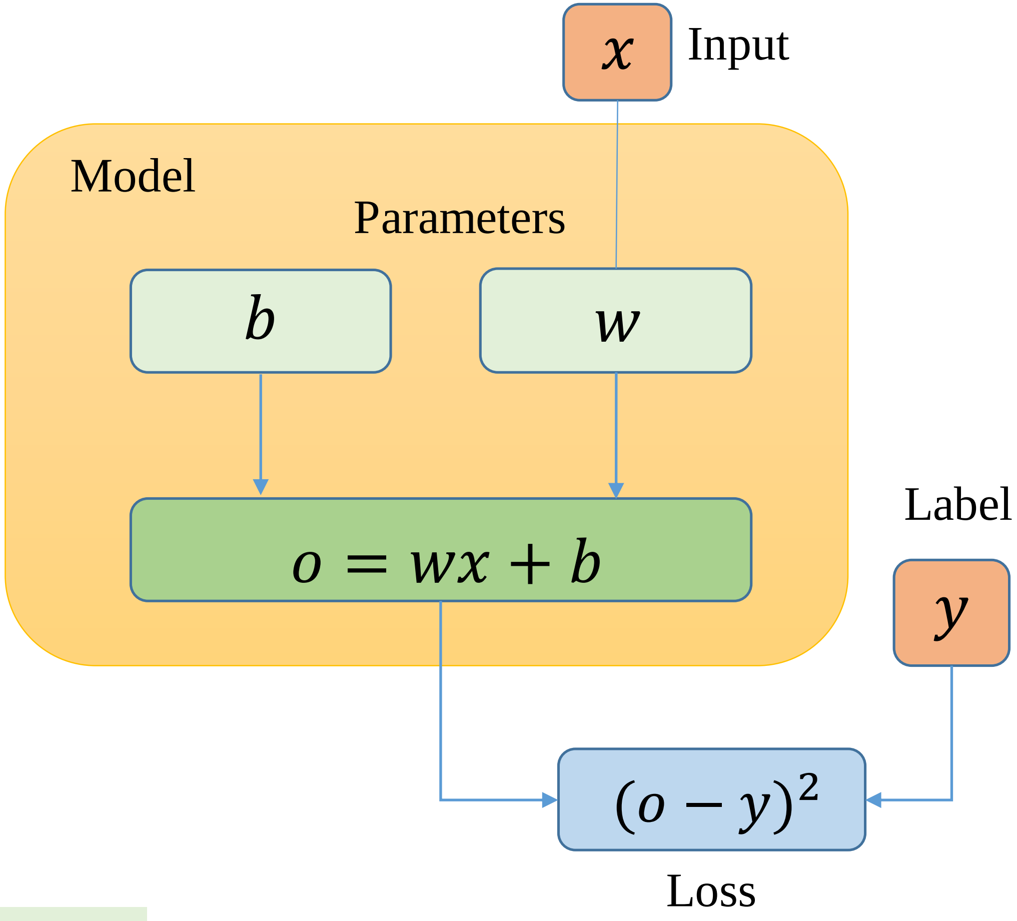
$$w_j = w_j - \eta L'_{w_j}$$

$$b = b - \eta L'_b$$

η is learning rate

Linear Regression

Diagram



Cheat sheet

Tính output o

$$o = \boldsymbol{wx} + b$$

Tính Loss

$$L = (o - y)^2$$

Tính đạo hàm

$$L'_{w_j} = 2x_j(o - y)$$

$$L'_b = 2(o - y)$$

Cập nhật tham số

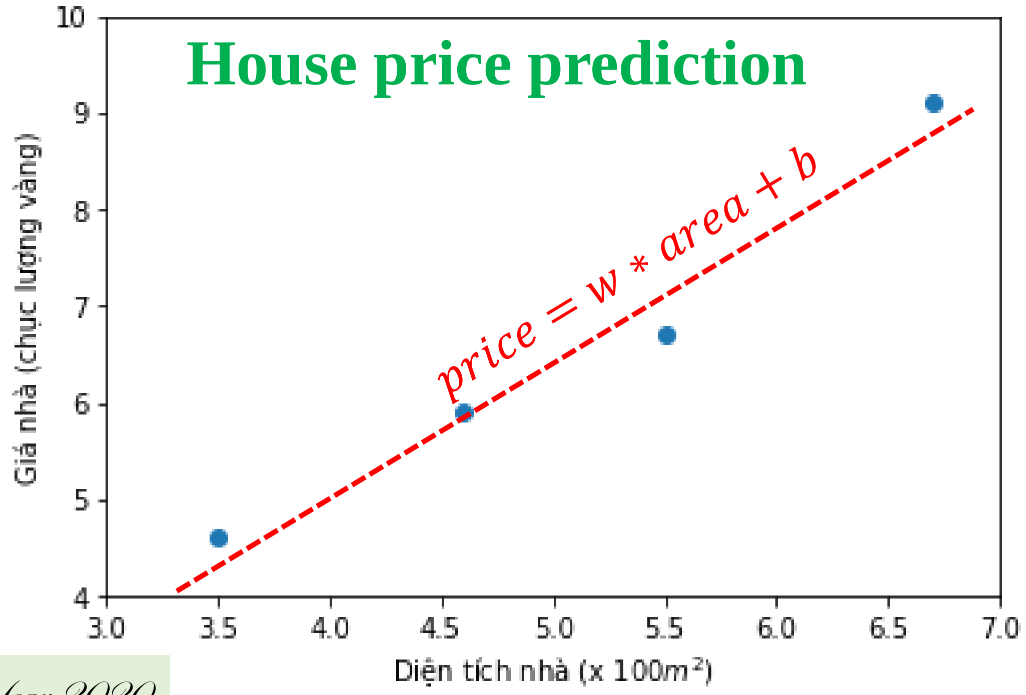
$$w_j = w_j - \eta L'_{w_j}$$

$$b = b - \eta L'_b$$

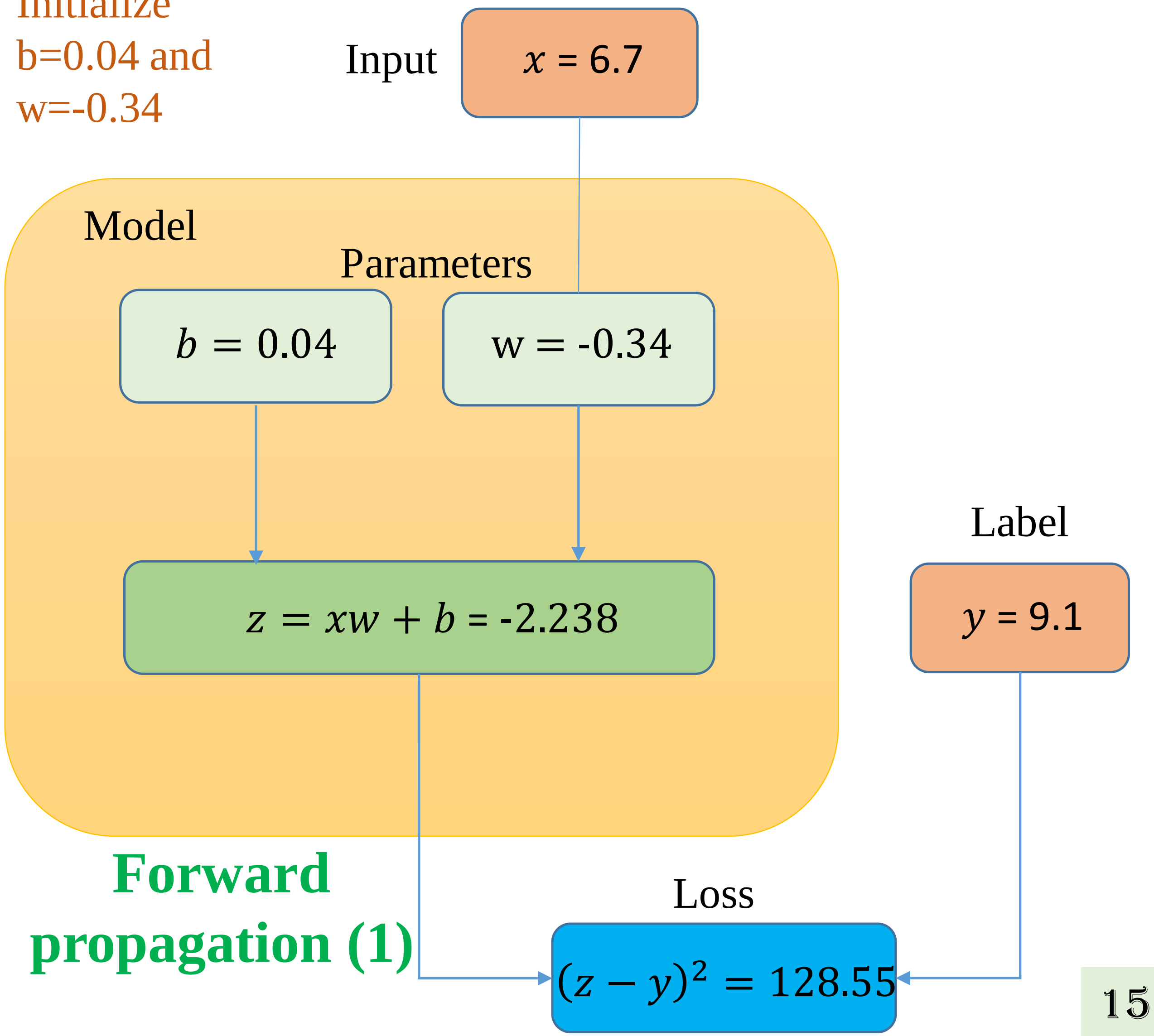
Linear Regression

Given sample data

	area	price	
	6.7	9.1	
	4.6	5.9	
	3.5	4.6	
	5.5	6.7	



Initialize
b=0.04 and
w=-0.34



Input

x = 6.7

Model

Parameters

b = 0.04

w = -0.34

$z = xw + b$ = -2.238

Label

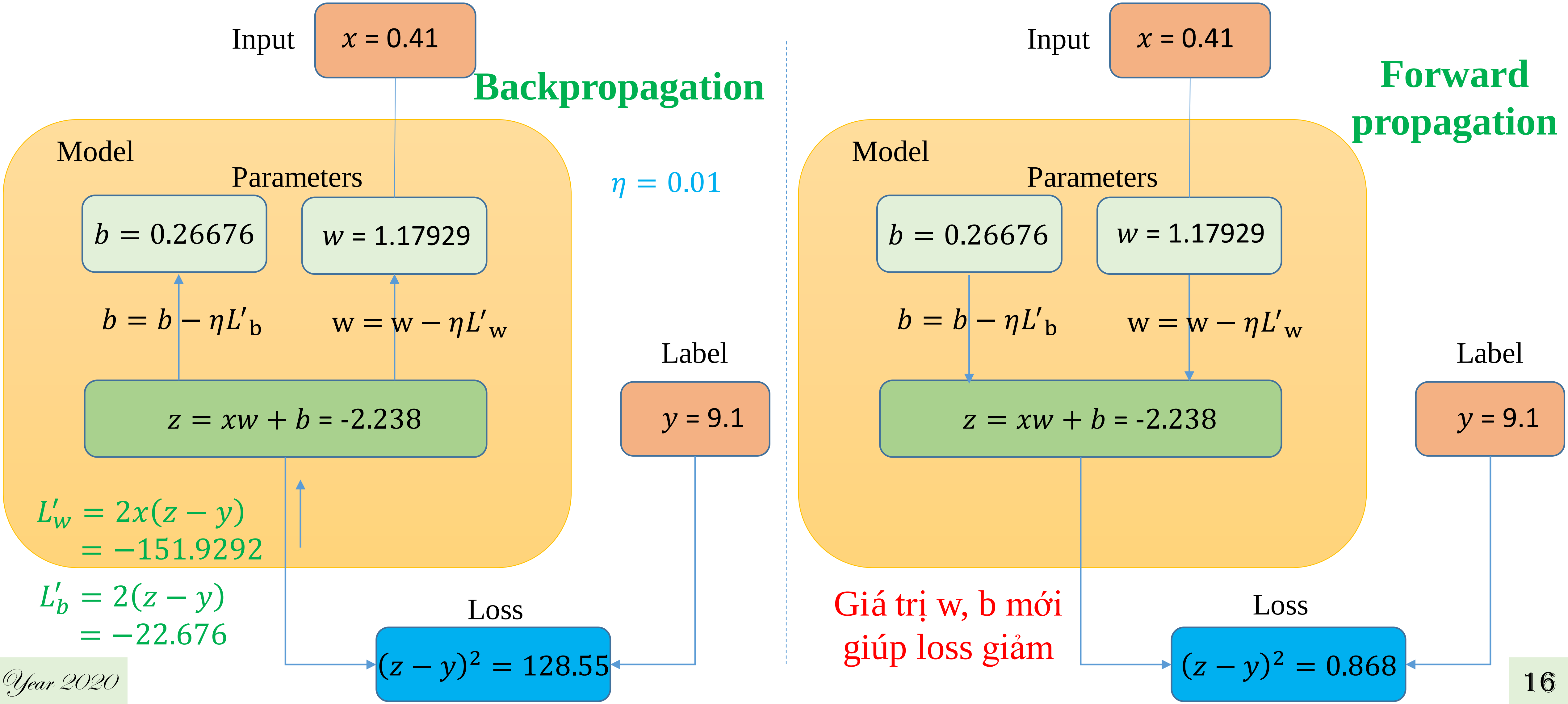
y = 9.1

Loss

$(z-y)^2 = 128.55$

Forward propagation (1)

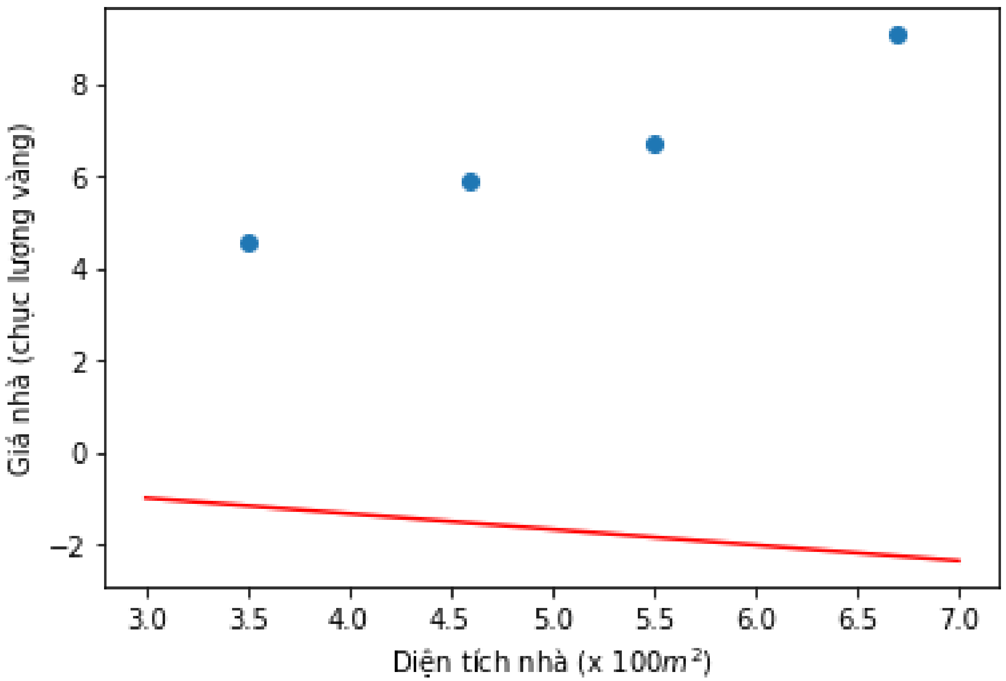
Linear Regression



Giá trị w, b mới giúp loss giảm

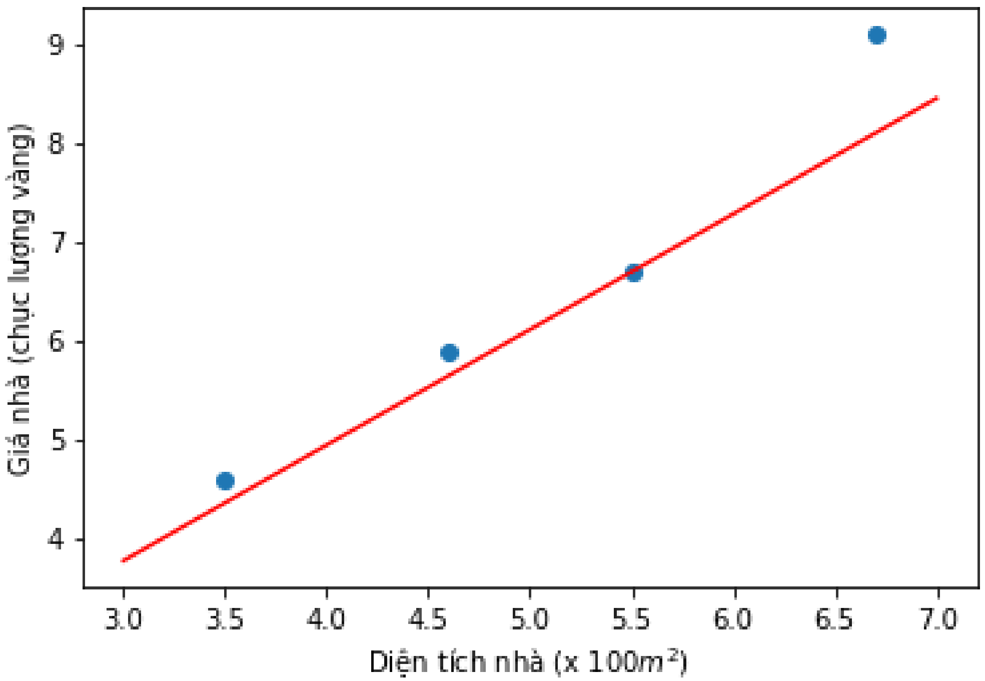
Linear Regression

Model prediction before and after the first update



w = -0.34 b = 0.04 L = 128.55

Before updating



w = 1.179292 b = 0.26676 L = 0.868

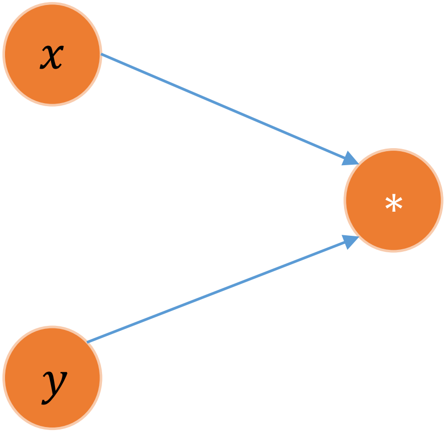
After updating

Outline

- ➢ Machine Learning
- ➢ Derivative/Gradient
- ➢ Linear Regression
- ➢ Computational Graph
- ➢ Generalized formula

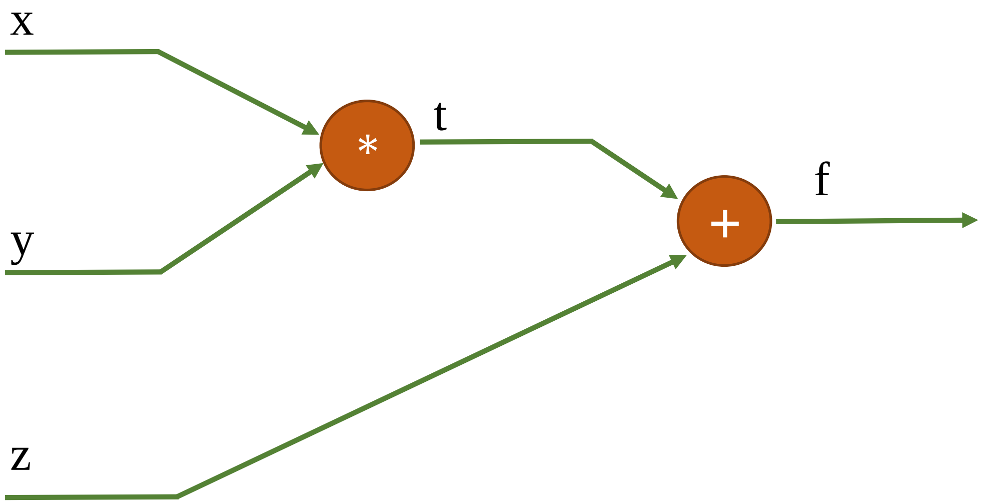
Computational graph

- A directed graph
- Nodes represent variables or operations



- Construct computational graph for $f(x, y, z) = x * y + z$
- Rewrite $f(x, y, z)$ as

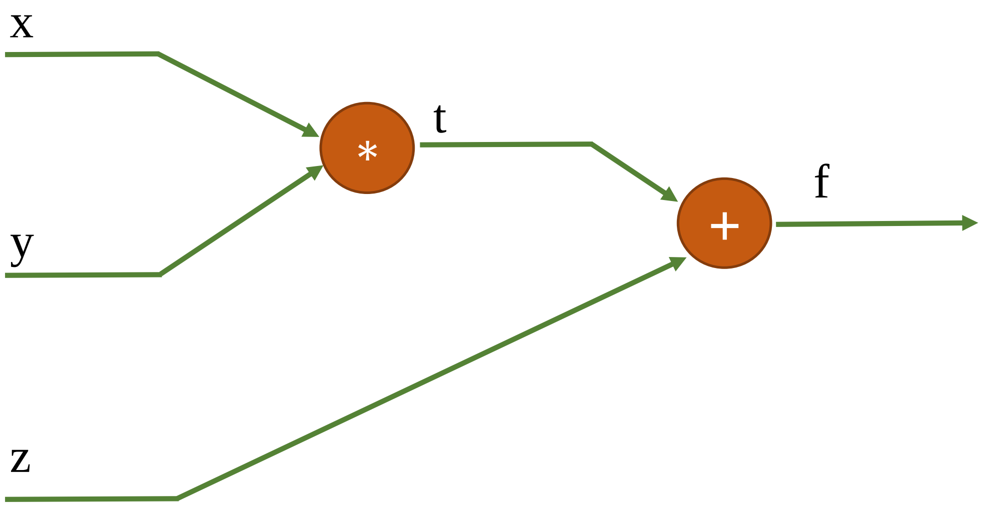
$f(t, z) = t + z$ where $t = x * y$



Computational graph

- Construct computational graph for $f(x, y, z) = x * y + z$
- Rewrite $f(x, y, z)$ as

$f(t, z) = t + z$ where $t = x * y$



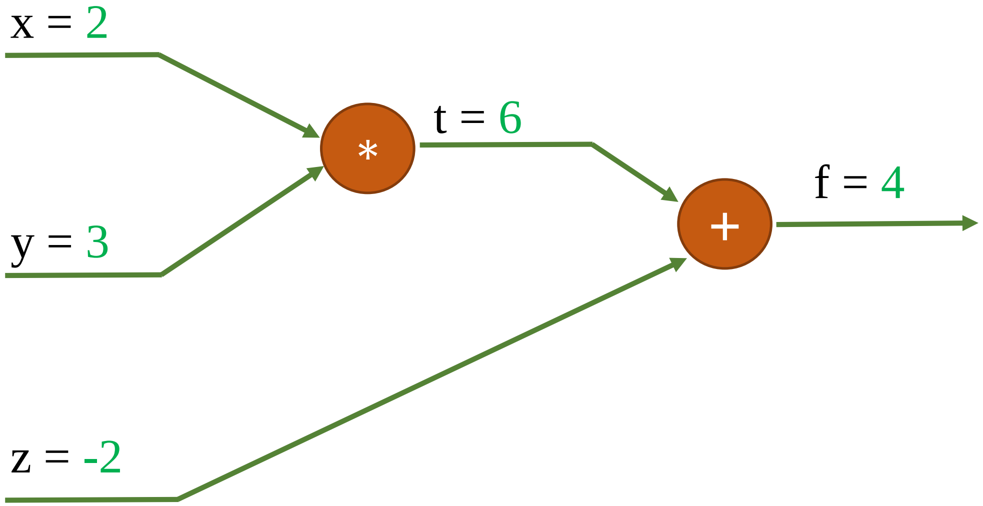
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

- **Compute** $f(x, y, z)$ **với** $x = 2$**,** $y = 3$ **và** $z = -2$**.**



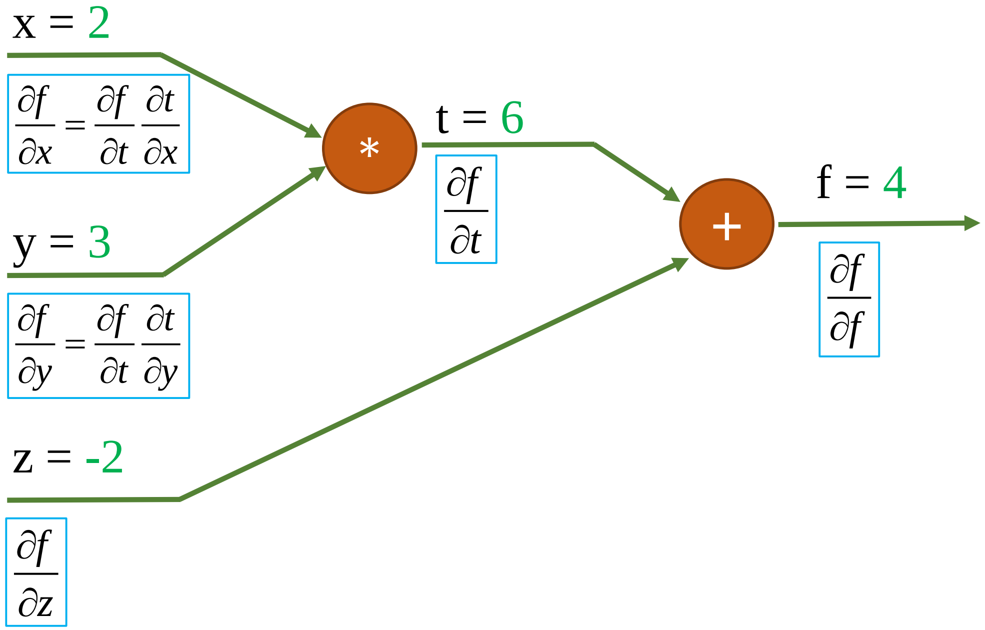
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

❖ **Compute** $f(x, y, z)$ **với** $x = 2$**,** $y = 3$ **và** $z = -2$**.**


x = 2
$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial t}\frac{\partial t}{\partial x}$
y = 3
$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial t}\frac{\partial t}{\partial y}$
z = -2
$\frac{\partial f}{\partial z}$
*
t = 6
$\frac{\partial f}{\partial t}$
+
f = 4
$\frac{\partial f}{\partial f}$

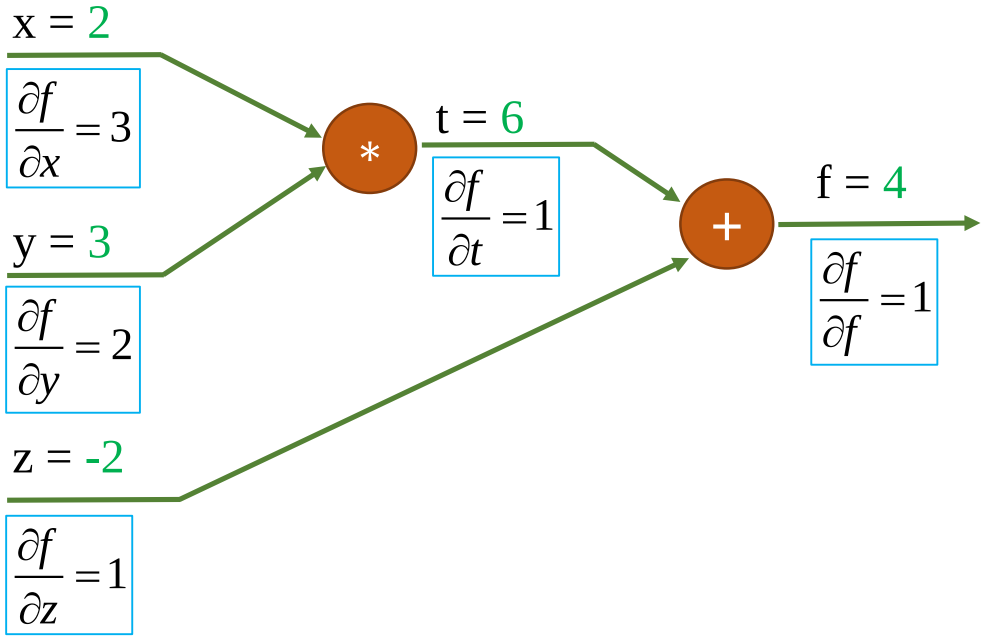
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

- ❖ **Compute** $f(x,y,z)$ **với** $x = 2$, $y = 3$ **và** $z = -2$.


x = 2
$\frac{\partial f}{\partial x} = 3$
y = 3
$\frac{\partial f}{\partial y} = 2$
z = -2
$\frac{\partial f}{\partial z} = 1$
*
t = 6
$\frac{\partial f}{\partial t} = 1$
+
f = 4
$\frac{\partial f}{\partial f} = 1$

Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

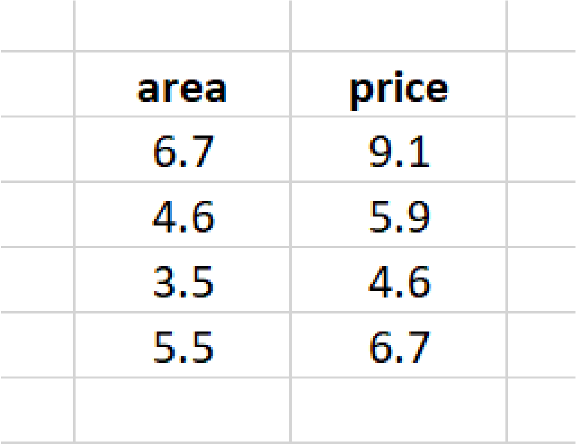
$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
  - 1-sample training
- Generalized formula

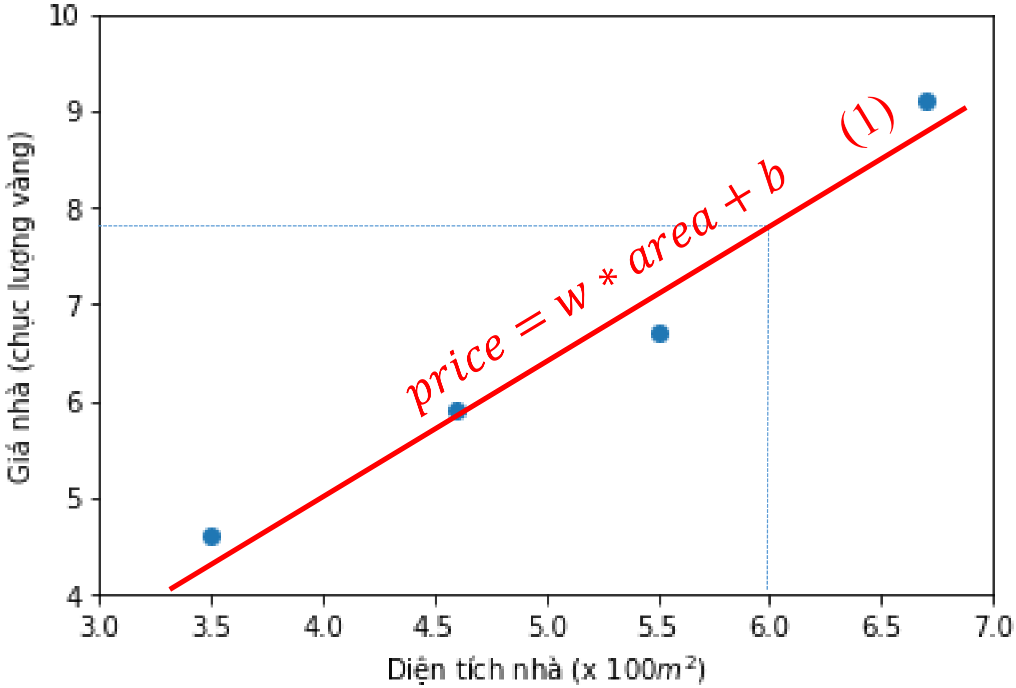
Computational graph

- **House price predictions**
  - **How much for a 600-m^2 house?**



	area	price
	6.7	9.1
	4.6	5.9
	3.5	4.6
	5.5	6.7

Given sample data



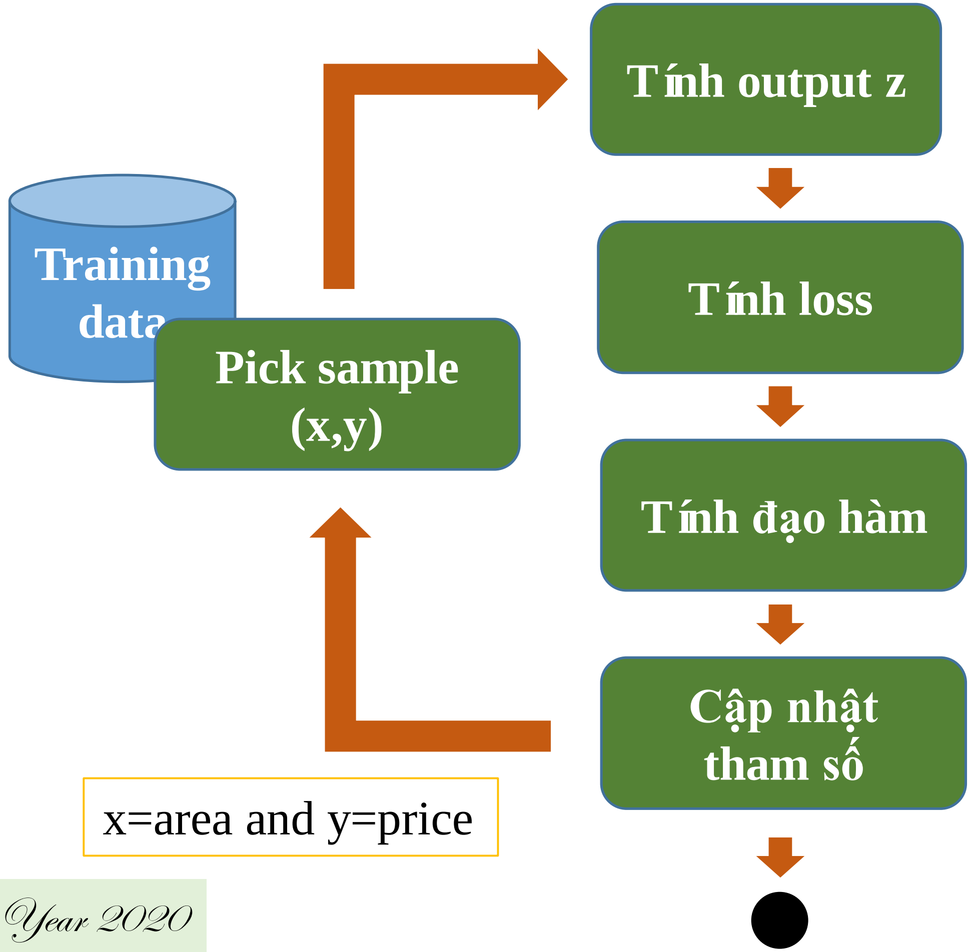
$price = w * area + b \quad (1)$

How to compute (1) using computational graph

Computational graph

❖ House price prediction

❖ One-sample training



x=area and y=price

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = wx + b$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_w = 2x(o - y)$$
$$L'_b = 2(o - y)$$

5) Cập nhật tham số

$$w = w - \eta L'_w$$
$$b = b - \eta L'_b$$

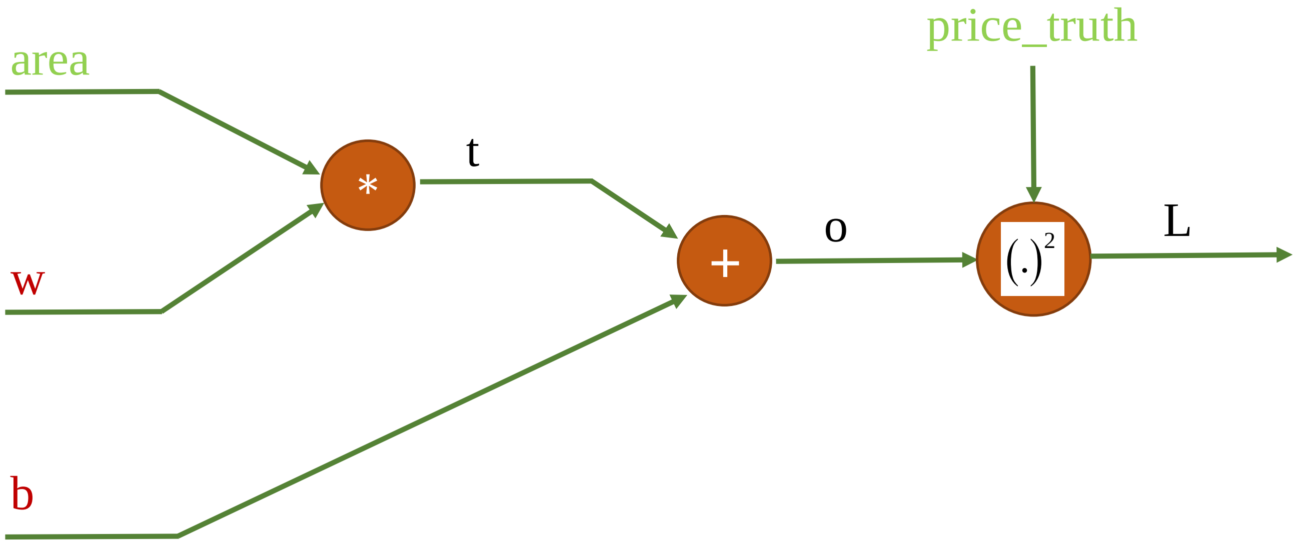
Learning rate η

Computational graph

- **House price prediction**
  - **One-sample training**

$$price = w * area + b$$

$$t = w * area$$

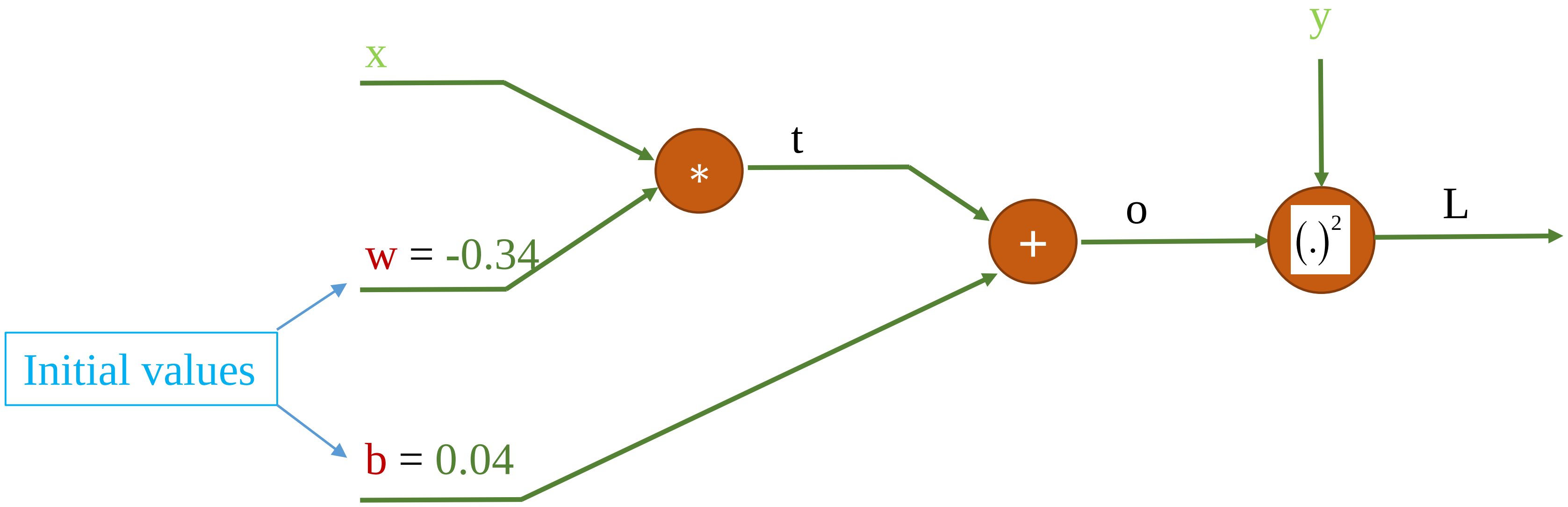


Computational graph

❖ House price prediction

❖ One-sample training

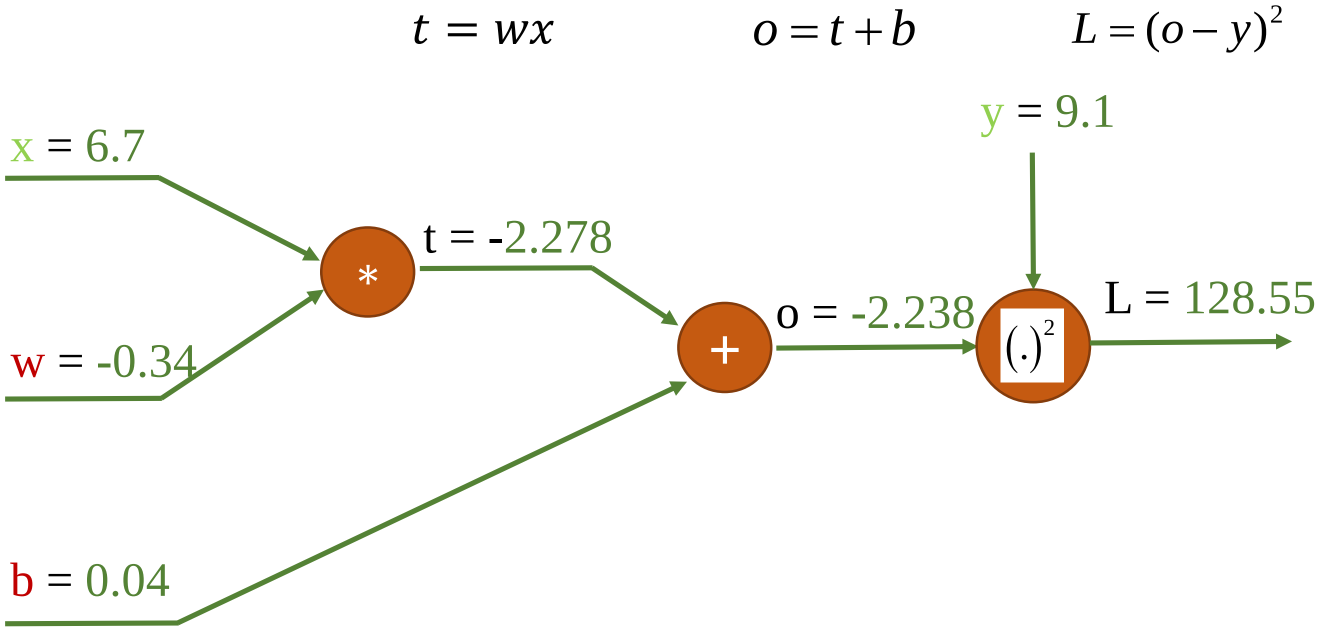
$t = wx$ $\quad o = t + b$ $\quad L = (o - y)^2$



Computational graph

❖ House price prediction

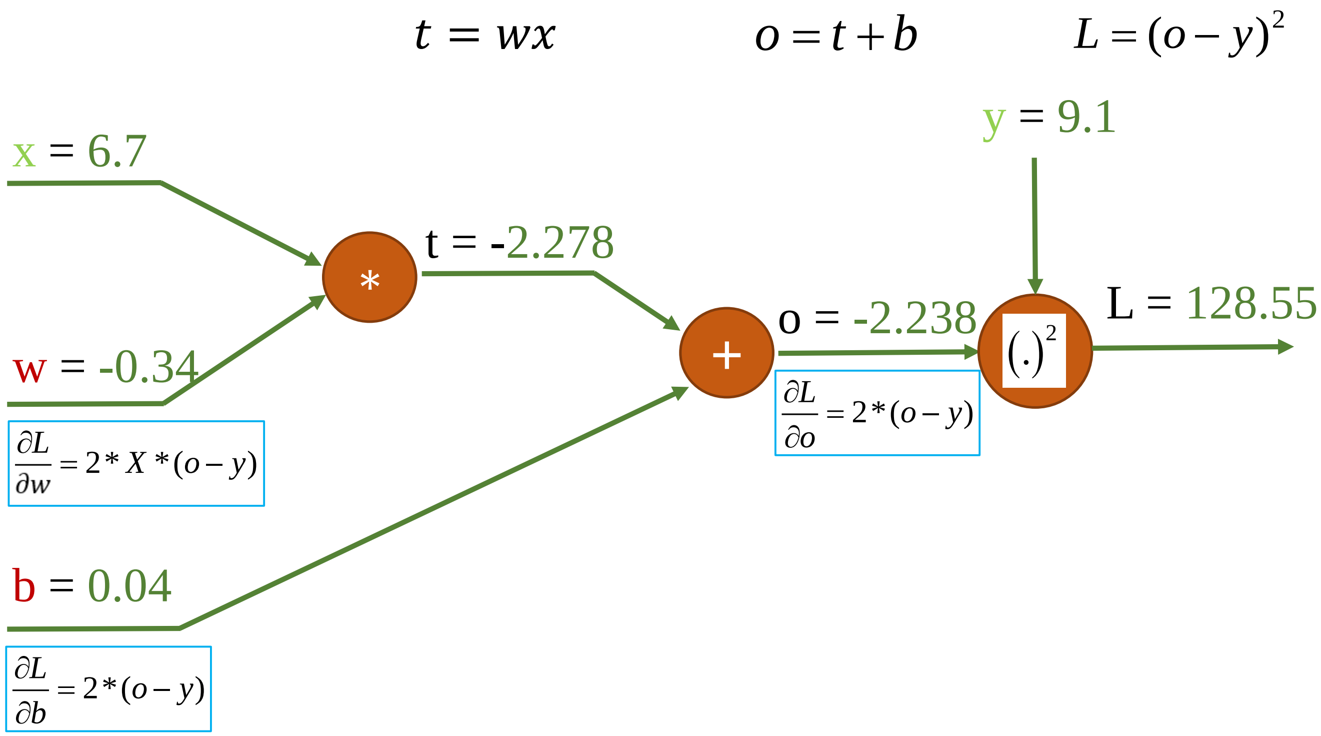
❖ One-sample training



Computational graph

❖ House price prediction

❖ One-sample training



$$\frac{\partial L}{\partial o} = 2*(o-y)$$

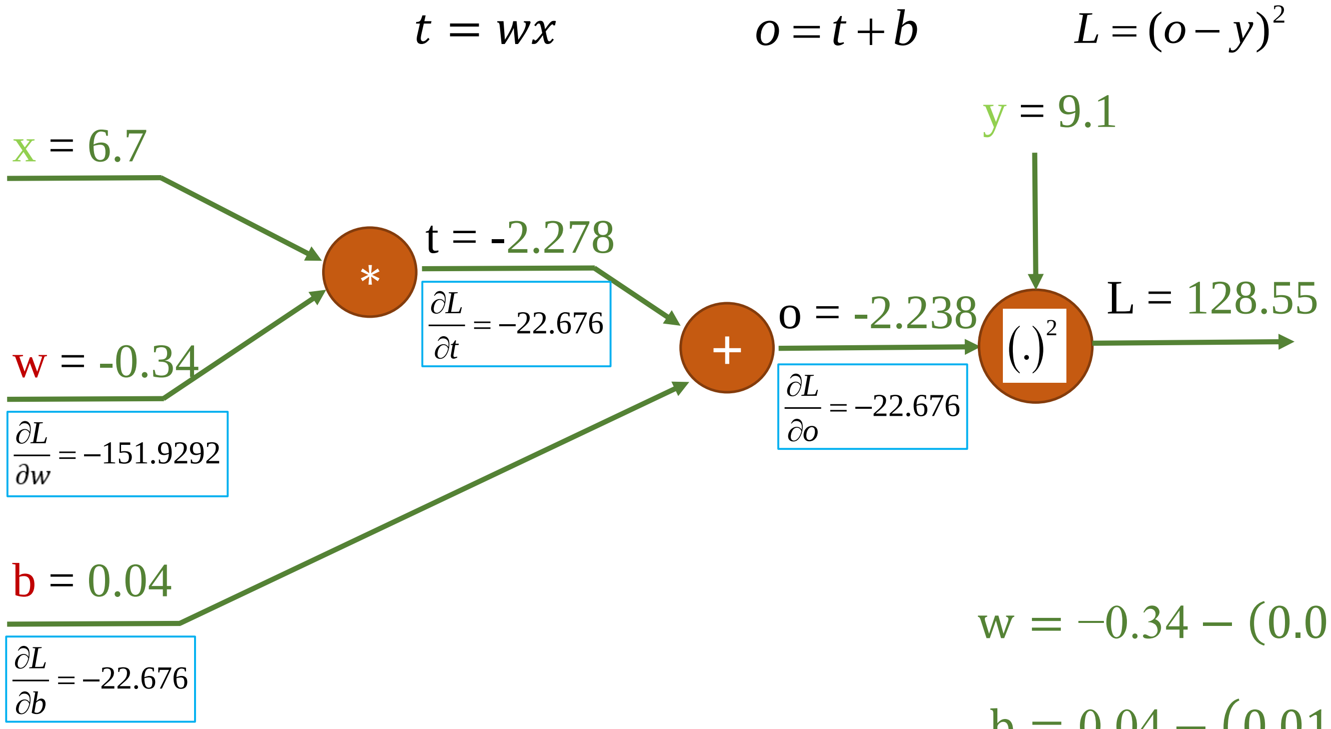
$$\frac{\partial L}{\partial w} = 2*X*(o-y)$$

$$\frac{\partial L}{\partial b} = 2*(o-y)$$

Computational graph

❖ House price prediction

❖ One-sample training


$t = wx$
$o = t + b$
$L = (o - y)^2$
x = 6.7
w = -0.34
b = 0.04
y = 9.1
*
+
$(.)^2$
t = -2.278
o = -2.238
L = 128.55
$\frac{\partial L}{\partial t} = -22.676$
$\frac{\partial L}{\partial o} = -22.676$
$\frac{\partial L}{\partial w} = -151.9292$
$\frac{\partial L}{\partial b} = -22.676$

Cách cập nhật a và b

$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

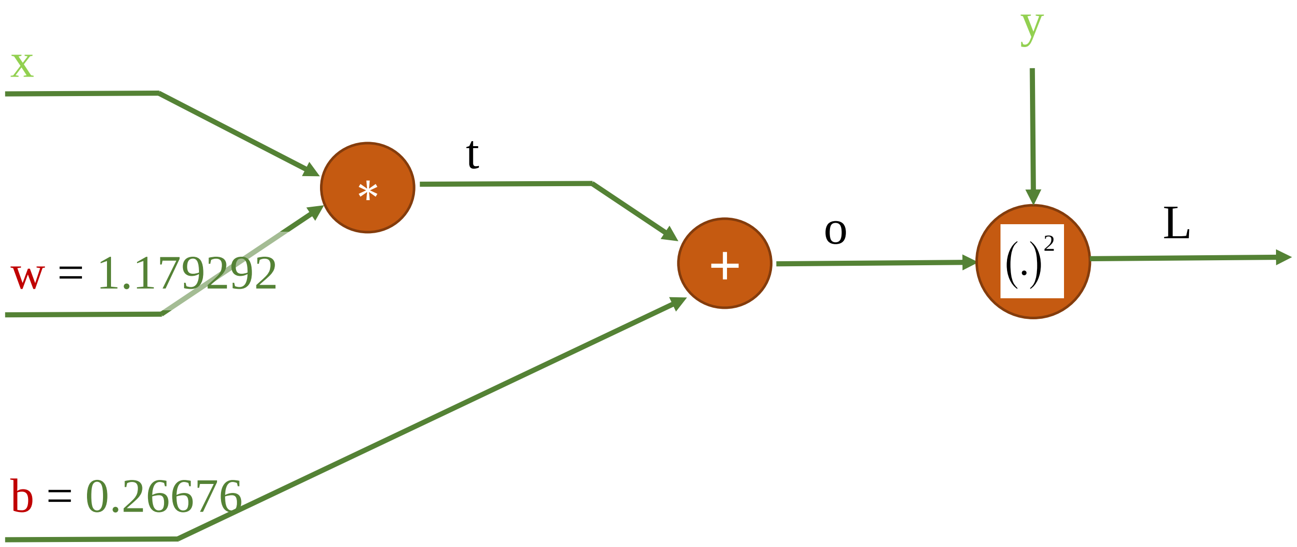
$$\text{w} = -0.34 - (0.01 * (-151.9)) = 1.179$$

$$\text{b} = 0.04 - \big(0.01 * (-22.67)\big) = 0.266$$

Computational graph

- **House price prediction**
  - **One-sample training**

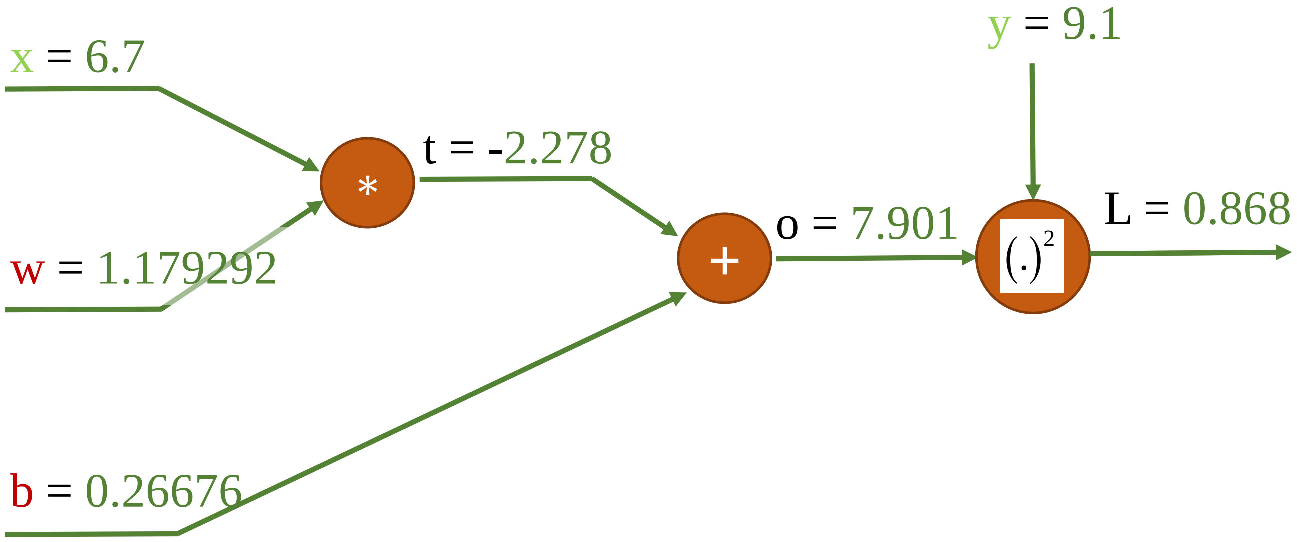
$t = wx$ $o = t + b$ $L = (o - y)^2$



Computational graph

- **House price prediction**
  - **One-sample training**

$t = wx$ $\quad o = t + b$ $\quad L = (o - y)^2$



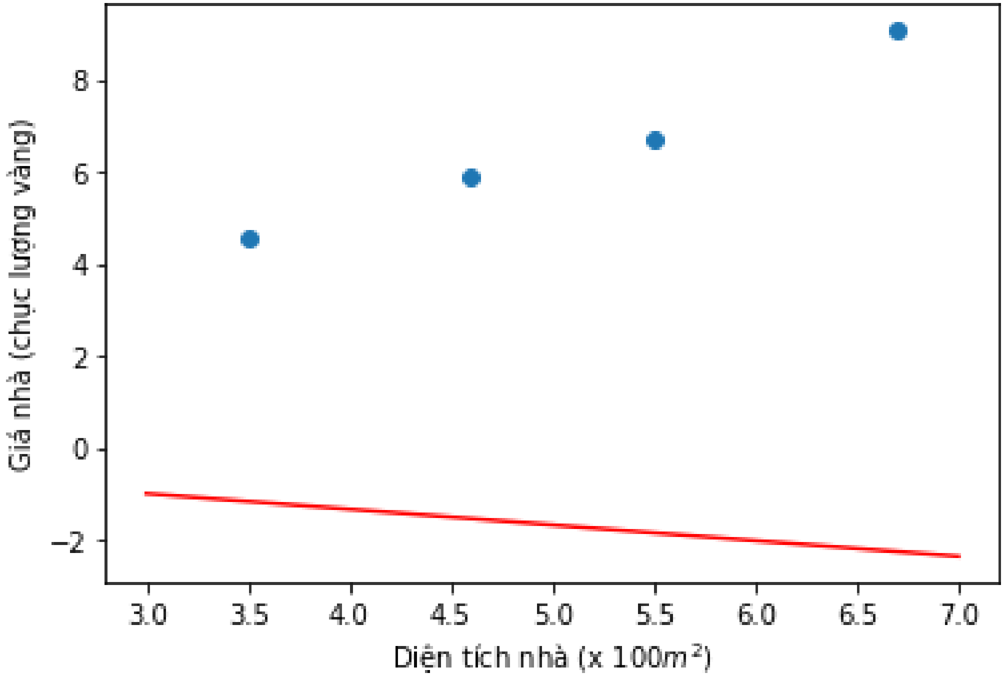
previous L = 128.55

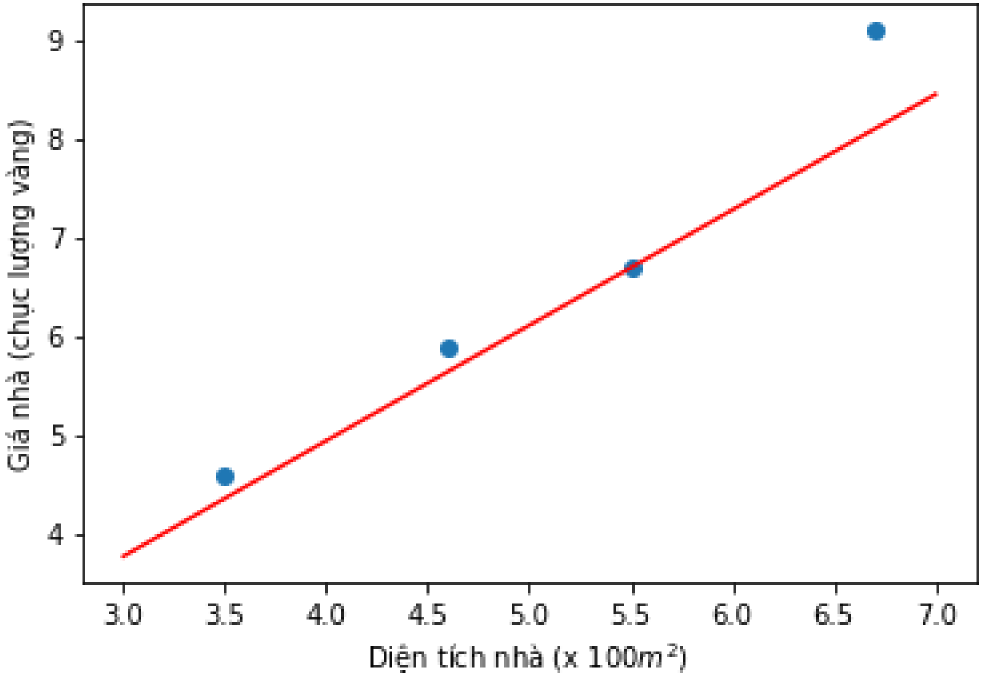
Updated a and b values help to reduce the L value

Computational graph

❖ House price prediction

❖ One-sample training



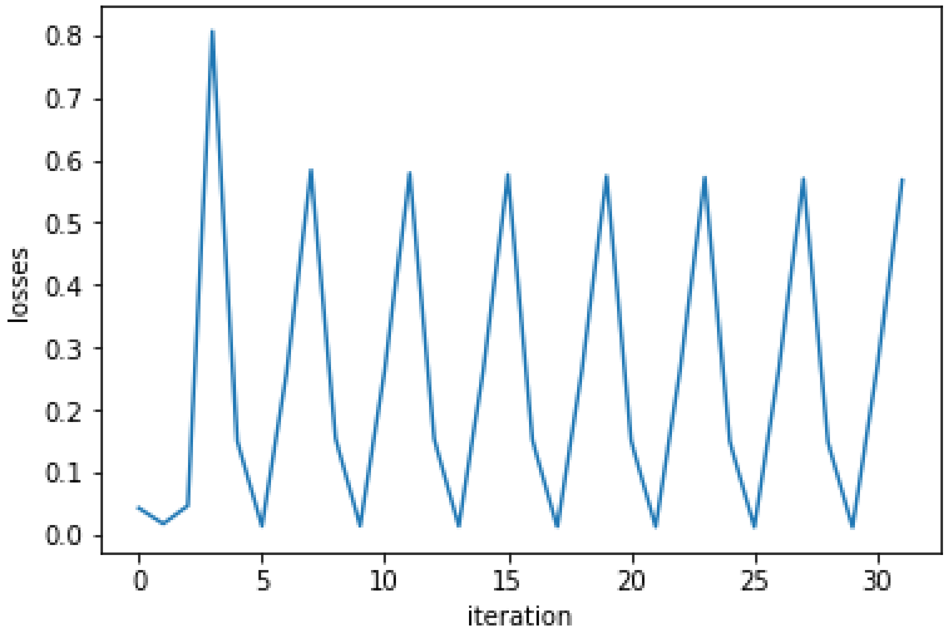


w = -0.34 b = 0.04 L = 128.55

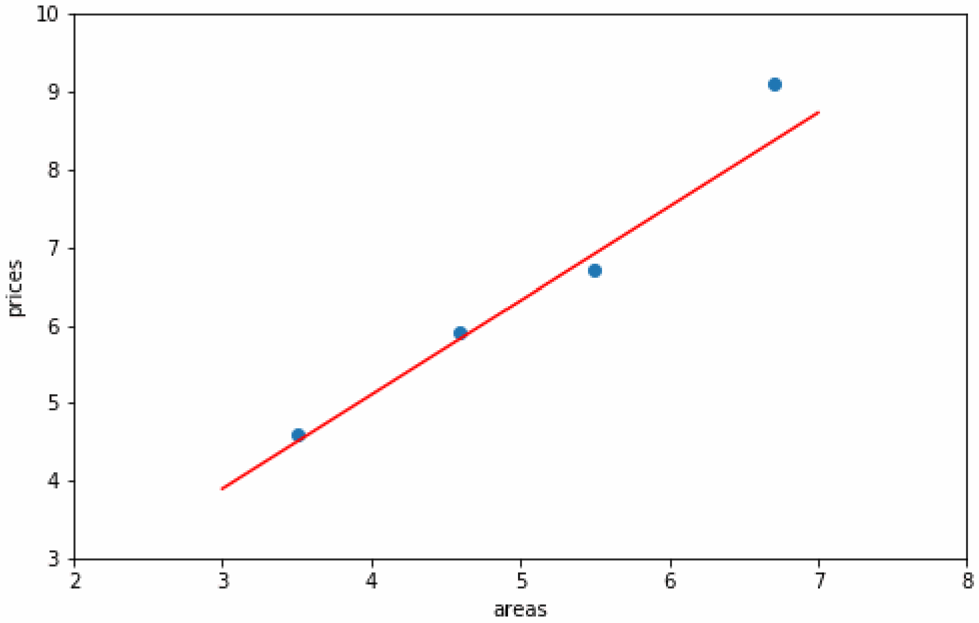
w = 1.179292 b = 0.26676 L = 0.868

Computational graph

- **House price prediction**
  - **One-sample training**



Losses for 30 iterations



Model updating for different iterations

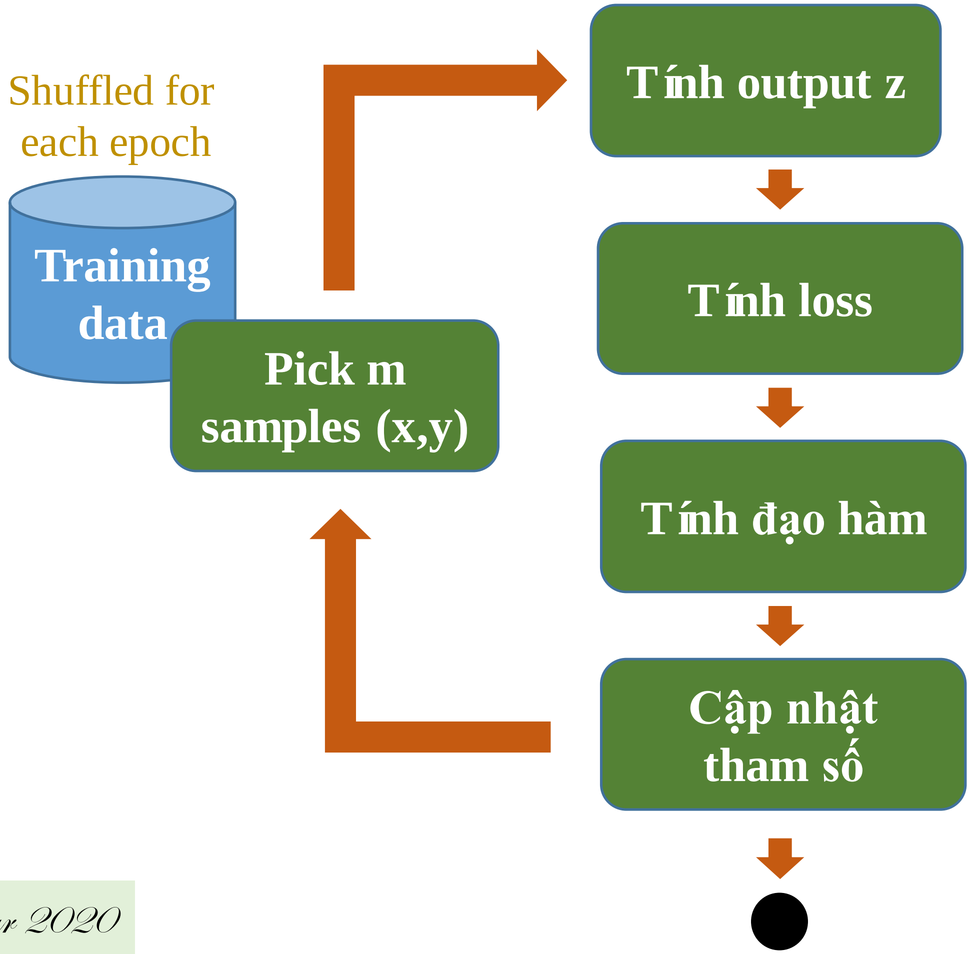
Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
    - 1-sample training
    - m-sample training
- Generalized formula

Computational graph

❖ House price prediction

❖ m-sample training (1<m<N)



1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

2) Tính output o_i

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < m$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

4) Tính đạo hàm

$$L_w'^{(i)} = 2x(o^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(o^{(i)} - y^{(i)}) \qquad \text{for } 0 \leq i < m$$

5) Cập nhật tham số

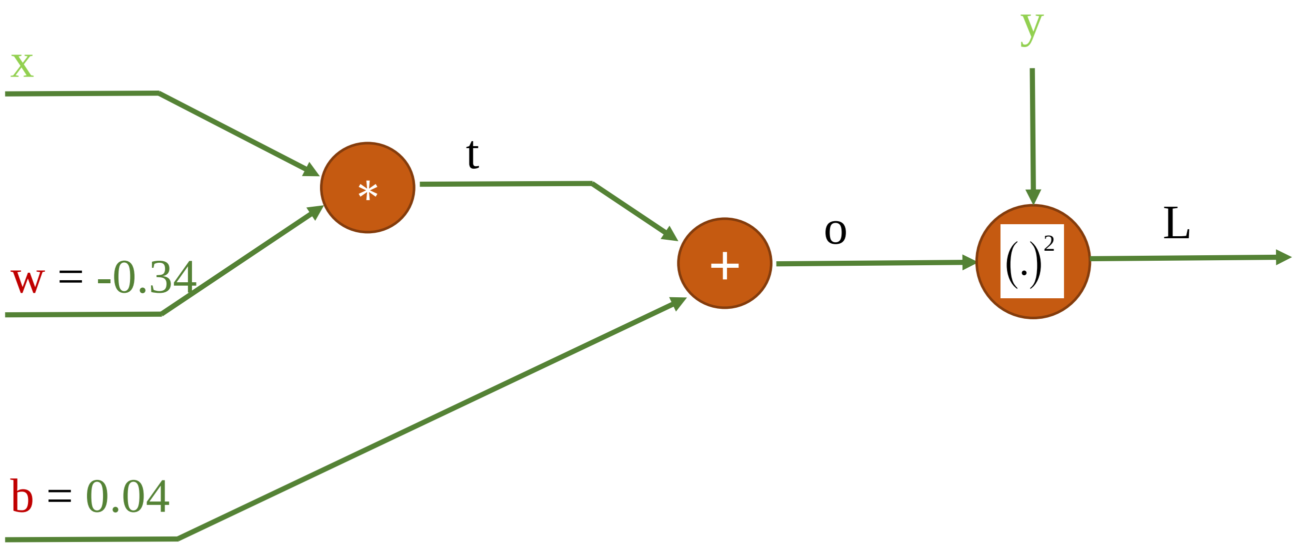
$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{m}$$

$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{m} \qquad \text{Learning rate } \eta$$

Computational graph

- **House price prediction**
  - **m-sample training (1<m<N)**

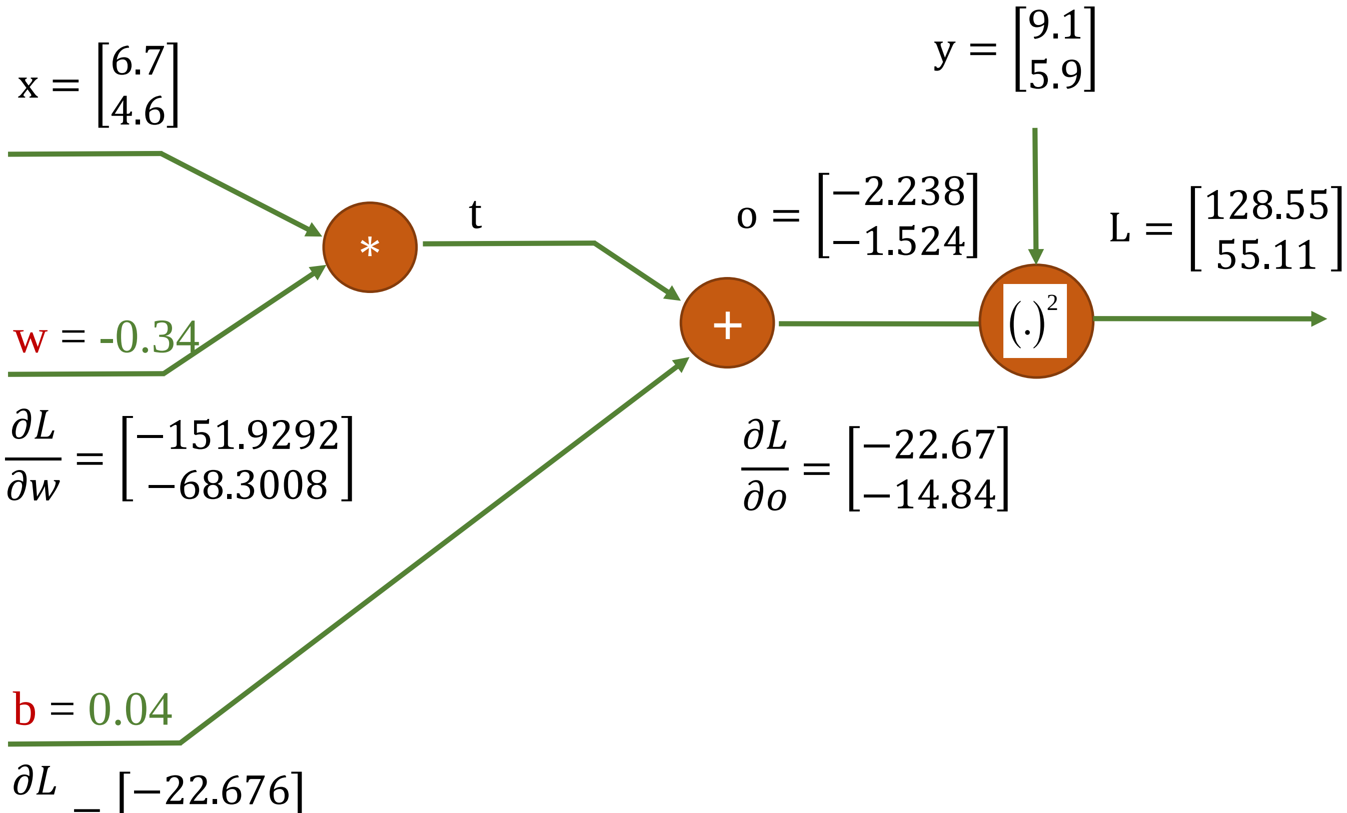
$t = wx$ $\qquad o = t + b$ $\qquad L = (o - y)^2$



Computational graph

- ❖ **House price prediction**
  - ❖ **m-sample training (1<m<N)**

m = 2



$x = \begin{bmatrix} 6.7 \\ 4.6 \end{bmatrix}$

$y = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$

$o = \begin{bmatrix} -2.238 \\ -1.524 \end{bmatrix}$

$L = \begin{bmatrix} 128.55 \\ 55.11 \end{bmatrix}$

w = -0.34

$\frac{\partial L}{\partial w} = \begin{bmatrix} -151.9292 \\ -68.3008 \end{bmatrix}$

$\frac{\partial L}{\partial o} = \begin{bmatrix} -22.67 \\ -14.84 \end{bmatrix}$

b = 0.04

$\frac{\partial L}{\partial b} = \begin{bmatrix} -22.676 \\ -14.848 \end{bmatrix}$

$$\frac{sum(\frac{\partial L}{\partial w})}{m} = -110.115$$

$$\frac{sum(\frac{\partial L}{\partial b})}{m} = -18.762$$

$t = wx$ $\quad$ $o = t + b$ $\quad$ $L = (o - y)^2$

Computational graph

❖ House price prediction

❖ m-sample training (1<m<N)

Cách cập nhật a và b

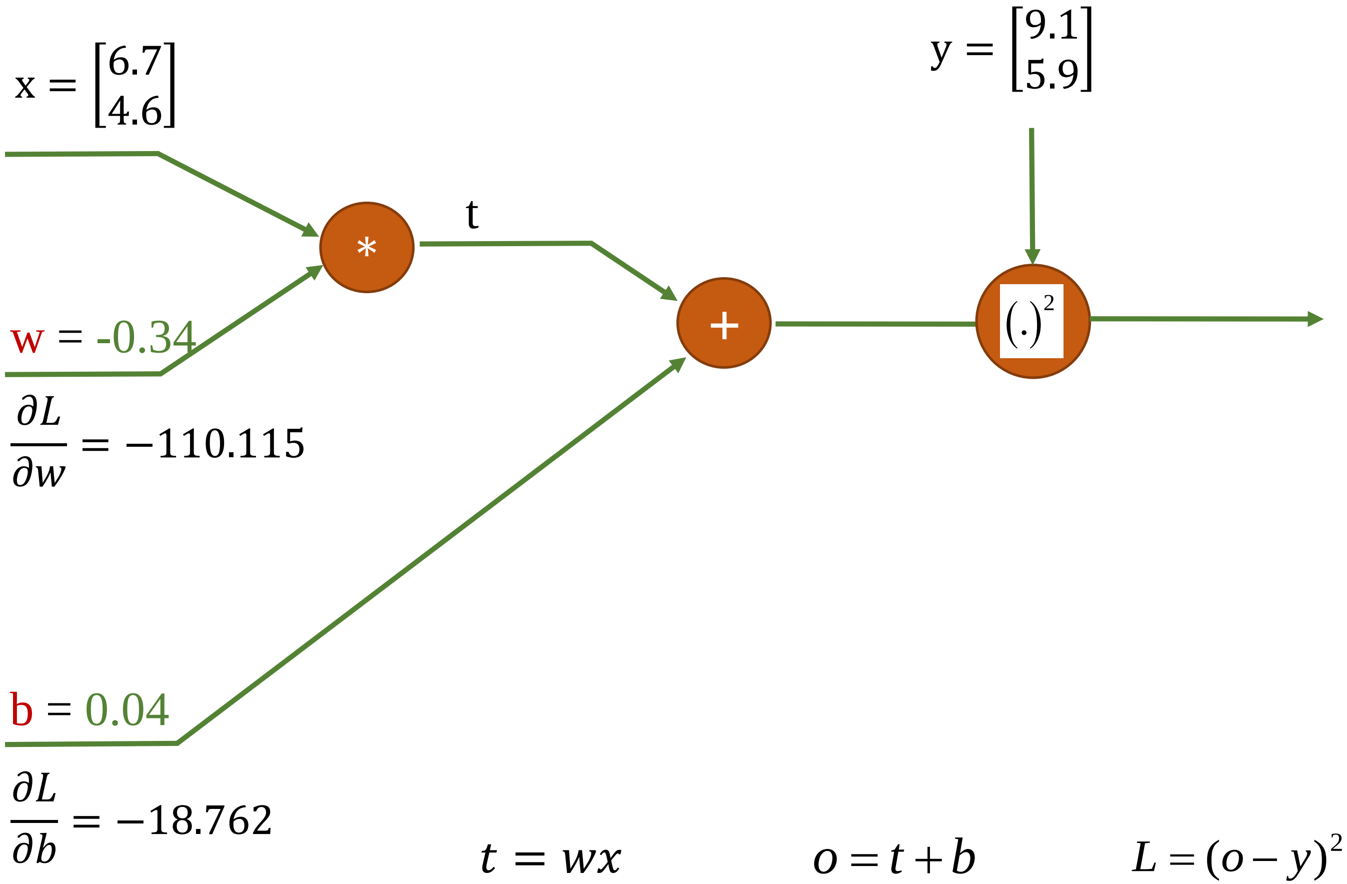
$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

$$\mathrm{w} = -0.34 - (0.01 * (-110.115)) = 0.761$$

$$\mathrm{b} = 0.04 - \left(0.01 * (-18.762)\right) = 0.227$$



$\mathrm{x} = \begin{bmatrix} 6.7 \\ 4.6 \end{bmatrix}$

$\mathrm{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$

w = -0.34

$\frac{\partial L}{\partial w} = -110.115$

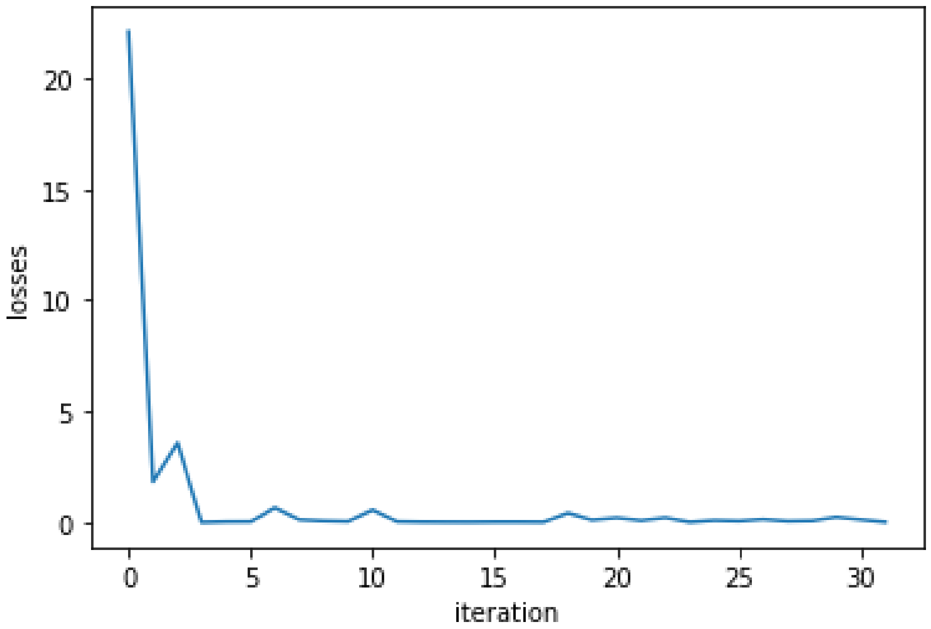
b = 0.04

$\frac{\partial L}{\partial b} = -18.762$

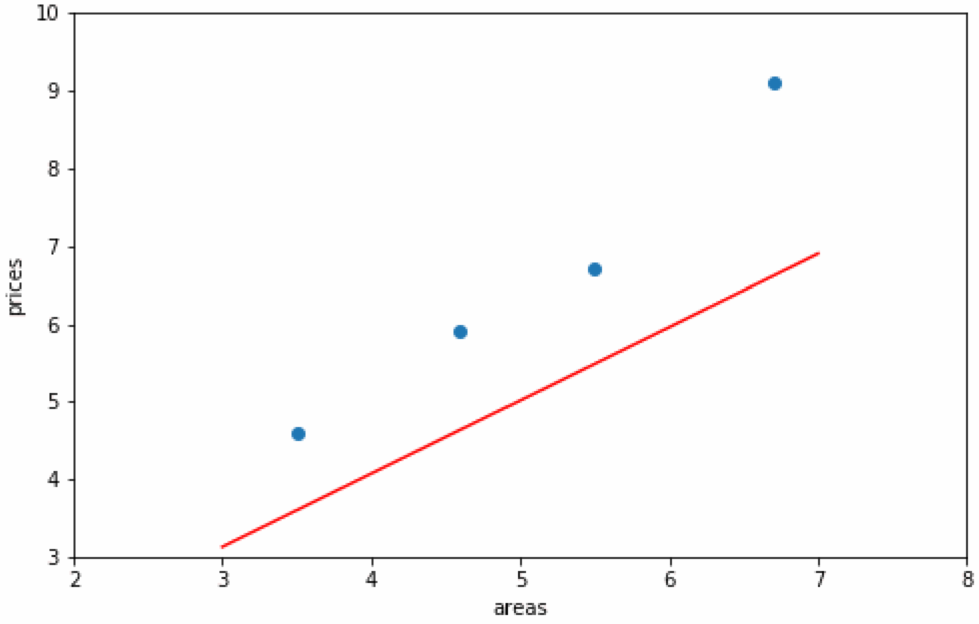
$t = wx$ $\quad o = t + b$ $\quad L = (o - y)^2$

Computational graph

- **House price prediction**
  - **m-sample training (1<m<N)**



Losses for 30 iterations



Model updating for different iterations

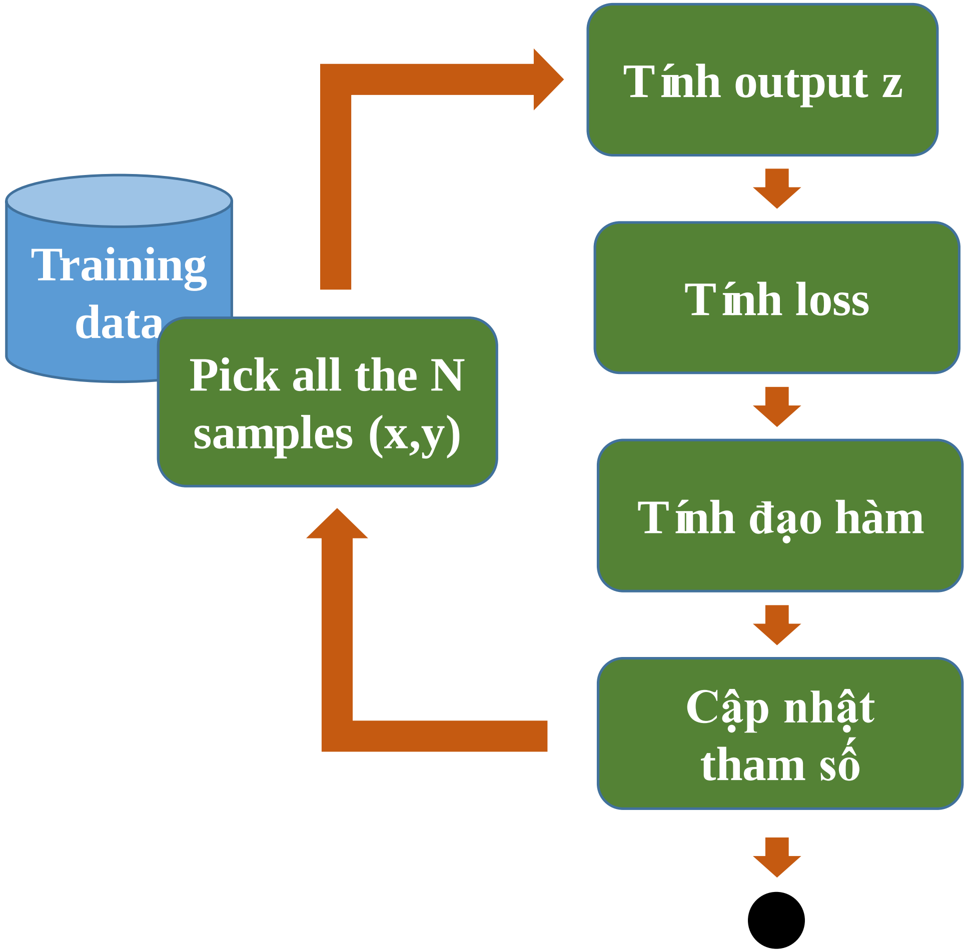
Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
    - 1-sample training
    - m-sample training
    - N-sample training
- Generalized formula

Computational graph

❖ House price prediction

❖ N-sample training



1) Pick all the N samples $(x^{(i)}, y^{(i)})$ from training data

2) Tính output o_i

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < N$$

4) Tính đạo hàm

$$L_w'^{(i)} = 2x(o^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < N$$

5) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{N}$$

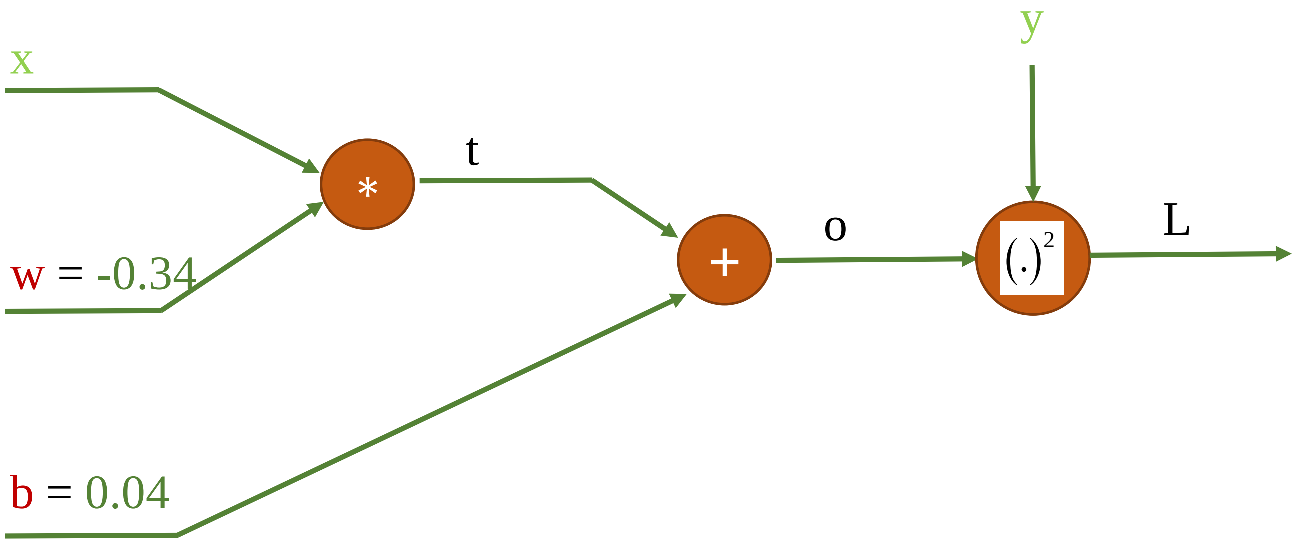
$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N} \qquad \text{Learning rate } \eta$$

Computational graph

❖ House price prediction

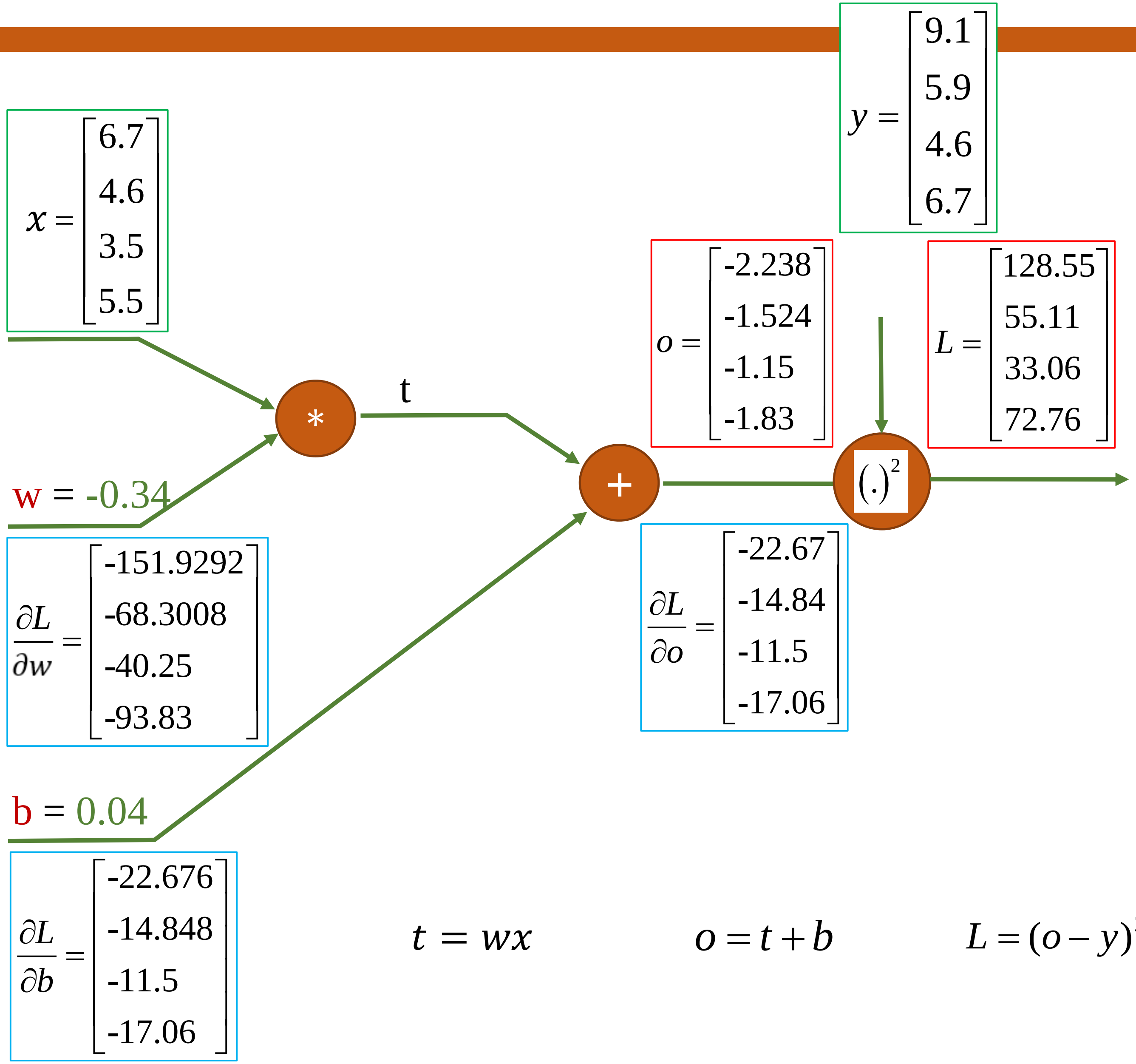
❖ N-sample training

$t = wx$ $o = t + b$ $L = (o - y)^2$



Computational graph

- **House price prediction**
  - **N-sample training**



$$x = \begin{bmatrix} 6.7 \\ 4.6 \\ 3.5 \\ 5.5 \end{bmatrix}$$

$$y = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

$$o = \begin{bmatrix} -2.238 \\ -1.524 \\ -1.15 \\ -1.83 \end{bmatrix}$$

$$L = \begin{bmatrix} 128.55 \\ 55.11 \\ 33.06 \\ 72.76 \end{bmatrix}$$

w = -0.34

b = 0.04

$$\frac{\partial L}{\partial w} = \begin{bmatrix} -151.9292 \\ -68.3008 \\ -40.25 \\ -93.83 \end{bmatrix}$$

$$\frac{\partial L}{\partial o} = \begin{bmatrix} -22.67 \\ -14.84 \\ -11.5 \\ -17.06 \end{bmatrix}$$

$$\frac{\partial L}{\partial b} = \begin{bmatrix} -22.676 \\ -14.848 \\ -11.5 \\ -17.06 \end{bmatrix}$$

$$\frac{sum(\frac{\partial L}{\partial w})}{4} = -88.5775$$

$$\frac{sum(\frac{\partial L}{\partial b})}{4} = -16.521$$

$t = wx$ $\quad$ $o = t + b$ $\quad$ $L = (o - y)^2$

Computational graph

- ❖ **House price prediction**
  - ❖ **N-sample training**

Cách cập nhật a và b

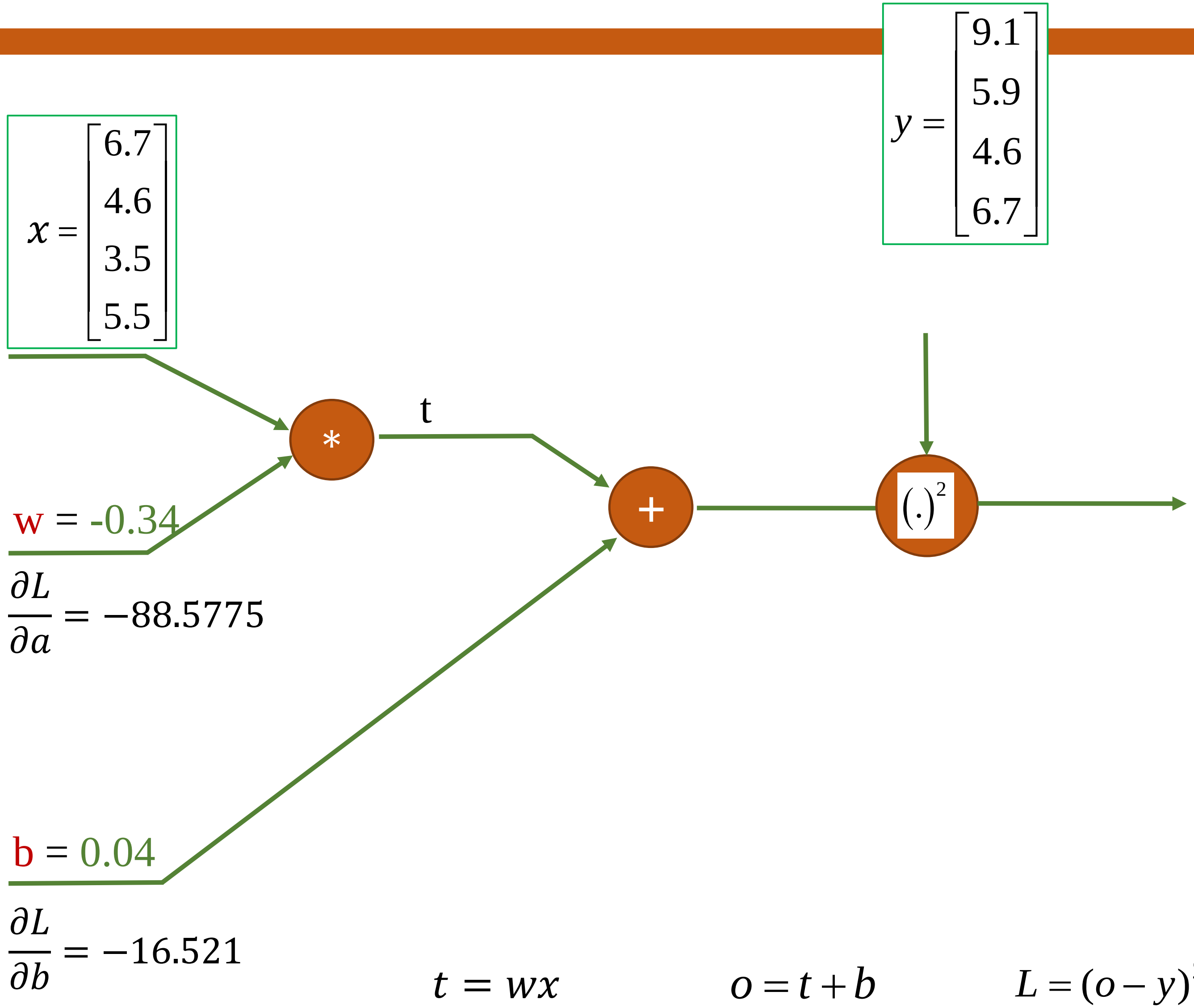
$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

$$w = -0.34 - (0.01 * (-88.5775)) = 0.54$$

$$b = 0.04 - \left(0.01 * (-16.521)\right) = 0.205$$



$$x = \begin{bmatrix} 6.7 \\ 4.6 \\ 3.5 \\ 5.5 \end{bmatrix}$$

$$y = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

w = -0.34

$$\frac{\partial L}{\partial a} = -88.5775$$

b = 0.04

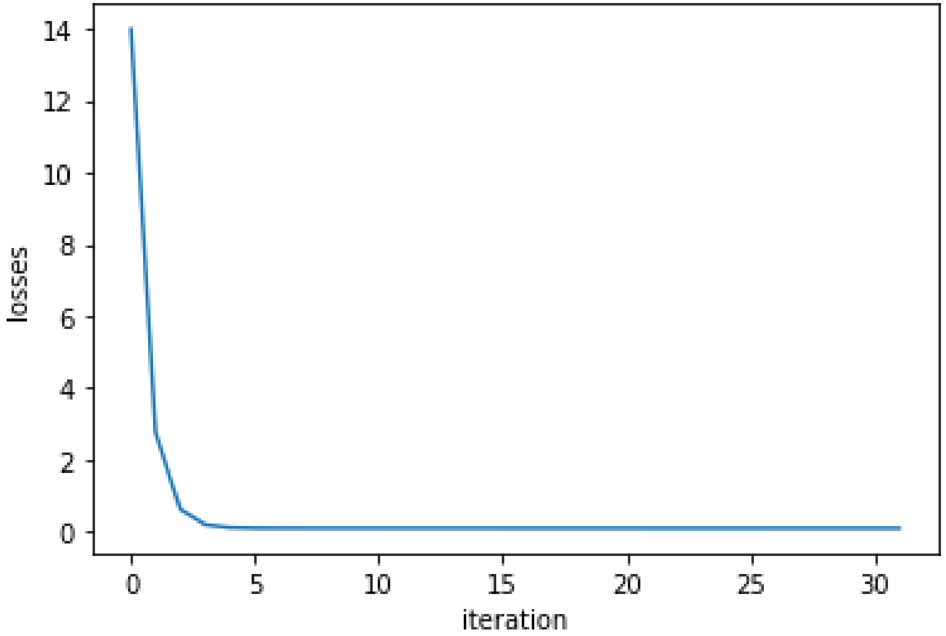
$$\frac{\partial L}{\partial b} = -16.521$$

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

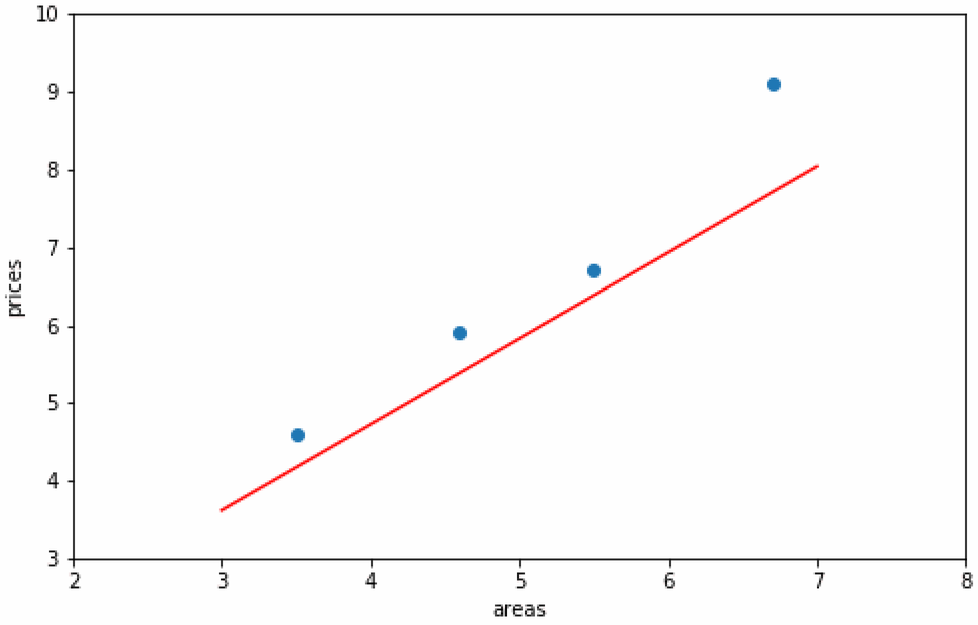
Computational graph

❖ House price prediction

❖ N-sample training



Losses for 30 iterations



Model updating for different iterations

Outline

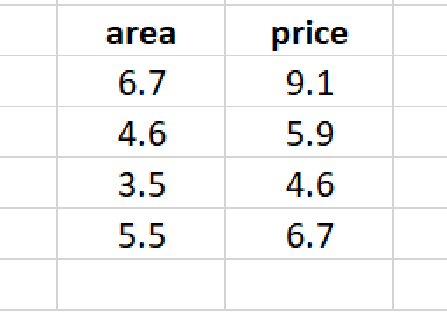
- ➢ Machine Learning
- ➢ Derivative/Gradient
- ➢ Linear Regression
- ➢ Computational Graph
- ➢ Generalized formula

Linear Regression

❖ Generalized formula

House price data

Feature Label



area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

Model

$$\text{price} = \text{w} * area + b$$

$$\text{y} = wx + b$$

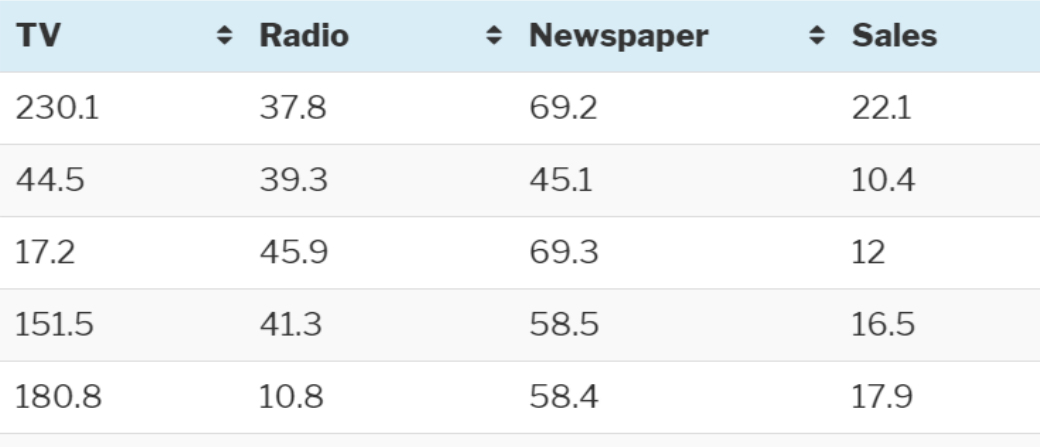
Model (vectorization)

$$\text{y} = \boldsymbol{\theta}^T\boldsymbol{x} \quad \text{where} \quad \boldsymbol{\theta}^T = [b \quad w]^T$$

$$\boldsymbol{x} = [x_0 \;\; area]^T$$

$$x_0 = 1$$

Features Label



TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Advertising data

Model

$$\text{Sale} = w_1 * TV + w_2 * Radio + w_3 * Newspaper + b$$

$$\text{y} = w_1x_1 + w_2x_2 + w_3x_3 + b$$

Model (vectorization)

$$\text{y} = \boldsymbol{\theta}^T\boldsymbol{x} \quad \text{where} \quad \boldsymbol{\theta}^T = [b \quad w_1 \quad w_2 \quad w_3]^T$$

$$\boldsymbol{x} = [x_0 \;\; TV \quad Radio \quad Newspaper]^T$$

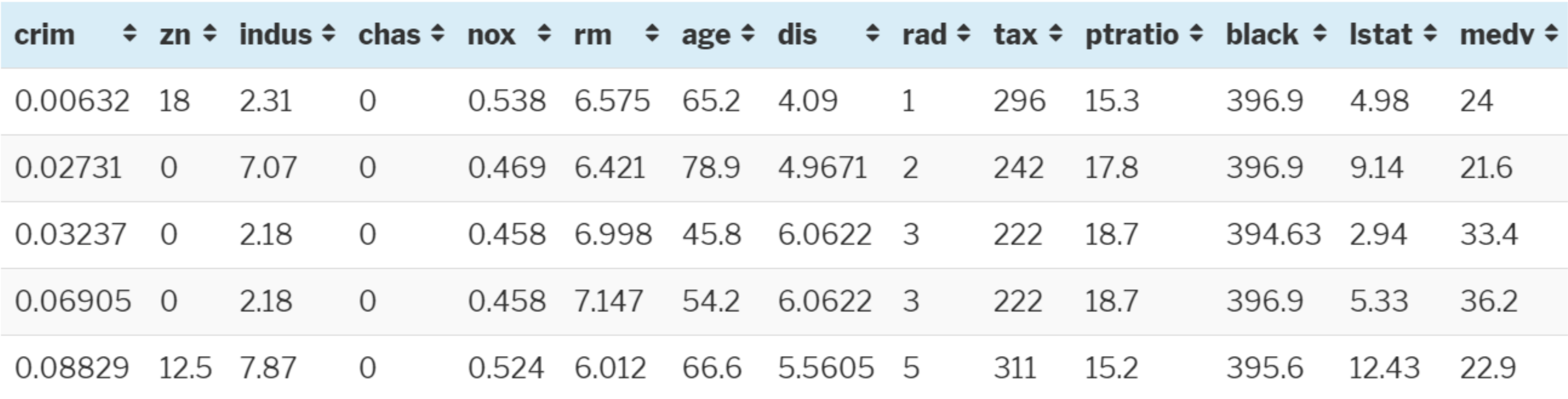
$$x_0 = 1$$

Linear Regression

❖ Generalized formula

Features | Label

Boston House Price Data



crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9

Model

$$\text{medv} = w_1 * x_1 + \cdots + w_{13} * x_{13} + b$$

Model (vectorization)

$$\text{y} = \boldsymbol{\theta}^T \boldsymbol{x} \qquad \text{where} \qquad \begin{aligned} \boldsymbol{\theta}^T &= [b \quad w_1 \quad \ldots \quad w_{13}]^T \\ \boldsymbol{x} &= [x_0 \quad x_1 \quad \ldots \quad x_{13}]^T \\ x_0 &= 1 \end{aligned}$$

Linear Regression (1-sample)

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = wx + b$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_w = 2x(o - y)$$

$$L'_b = 2(o - y)$$

5) Cập nhật tham số

$$w = w - \eta L'_w$$

$$b = b - \eta L'_b$$

η is learning rate

Friendly version

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = \boldsymbol{\theta}^T \boldsymbol{x}$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o - y)$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Generalized formula

Linear Regression (m-samples)

1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

1.1) Tính output $o^{(i)}$

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < m$$

1.2) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

1.3) Tính đạo hàm

$$L_w'^{(i)} = 2x(o^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

2) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{m}$$
$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{m}$$

η is learning rate

Friendly version

1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

1.1) Tính output $o^{(i)}$

$$o^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} \qquad \text{for } 0 \leq i < m$$

1.2) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \qquad \text{for } 0 \leq i < m$$

1.3) Tính đạo hàm

$$L_{\boldsymbol{\theta}}'^{(i)} = 2\boldsymbol{x}(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

2) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}'^{(i)}}{m}$$

η is learning rate

Generalized formula

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Tính output $o^{(i)}$

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < N$$

4) Tính đạo hàm

$$\left.\begin{aligned} L_w'^{(i)} &= 2x(o^{(i)} - y^{(i)}) \\ L_b'^{(i)} &= 2(o^{(i)} - y^{(i)}) \end{aligned}\right. \quad \text{for } 0 \leq i < N$$

5) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{N}$$

$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N}$$

η is learning rate

Friendly version

1) Pick all the N samples from training data

2) Tính output $o^{(i)}$

$$o^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} \qquad \text{for } 0 \leq i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \qquad \text{for } 0 \leq i < N$$

4) Tính đạo hàm

$$L_{\boldsymbol{\theta}}'^{(i)} = 2\boldsymbol{x}(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < N$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}'^{(i)}}{N}$$

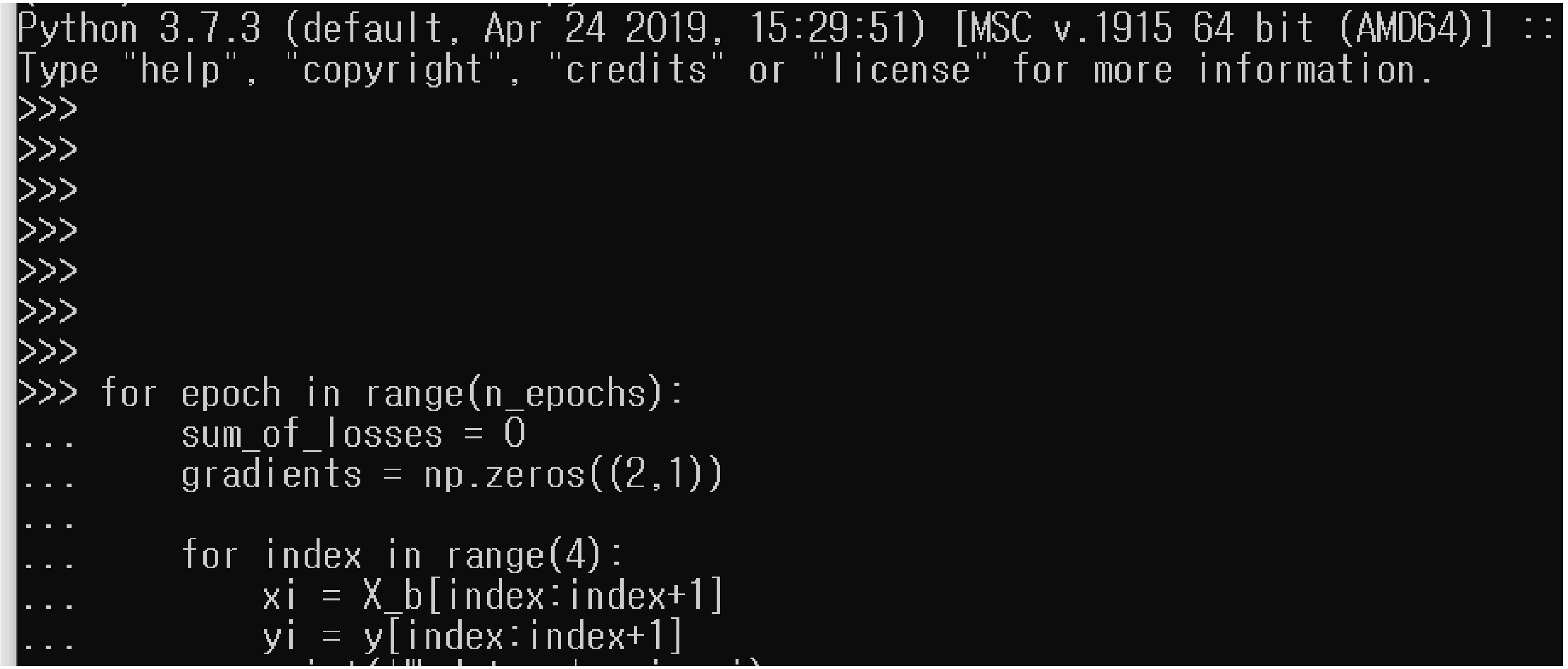
η is learning rate

Generalized formula

Linear Regression

❖ House price prediction

Demo

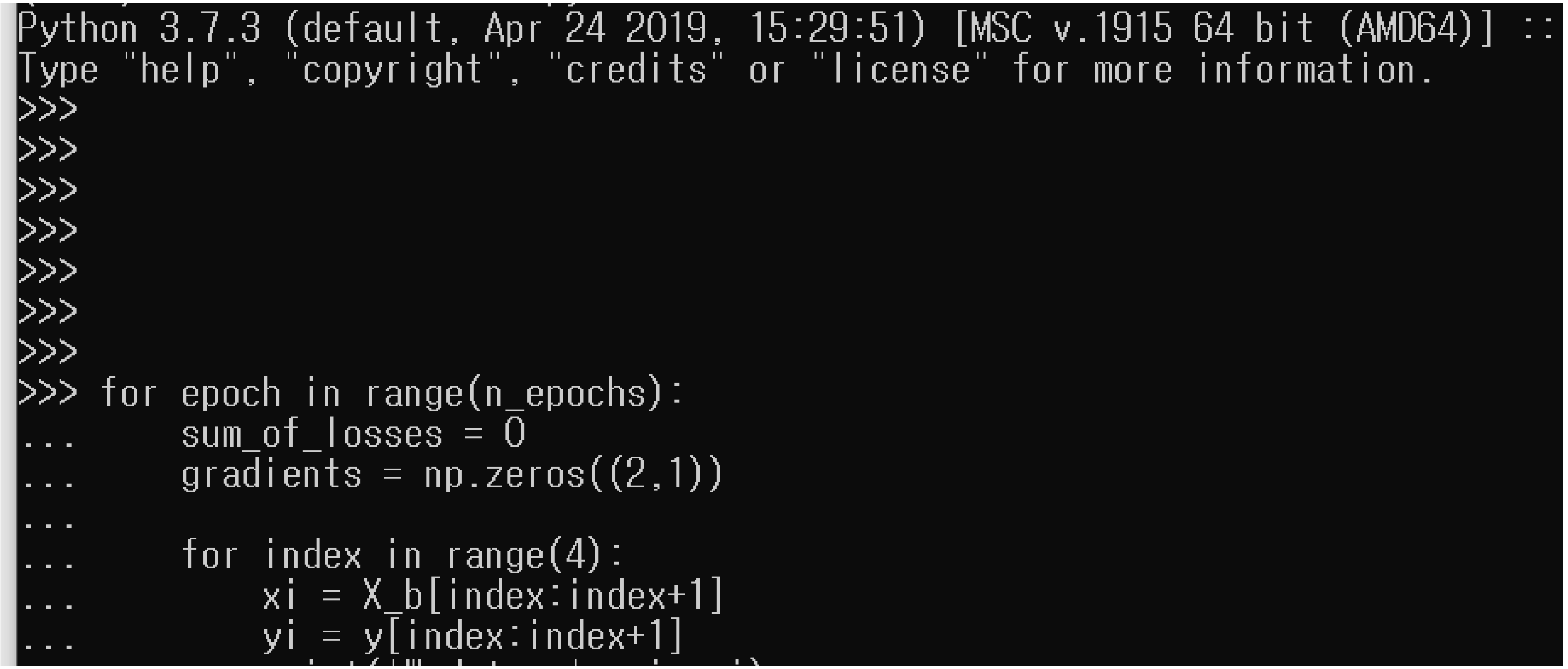


```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

Linear Regression

❖ Advertising-based sale prediction

Demo



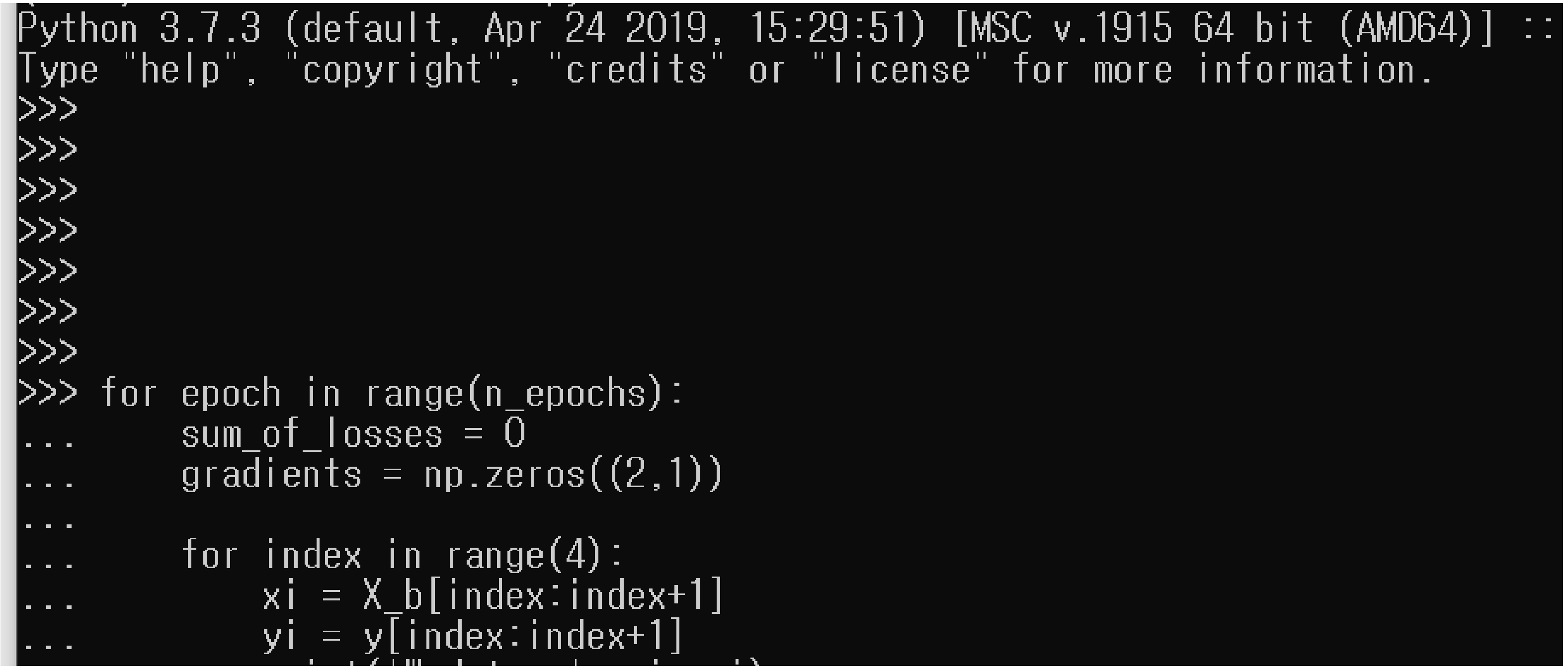
```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

Linear Regression

❖ House price prediction

❖Boston data

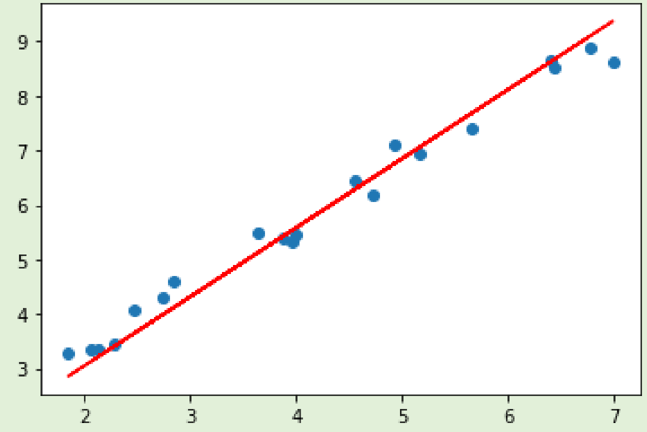
Demo



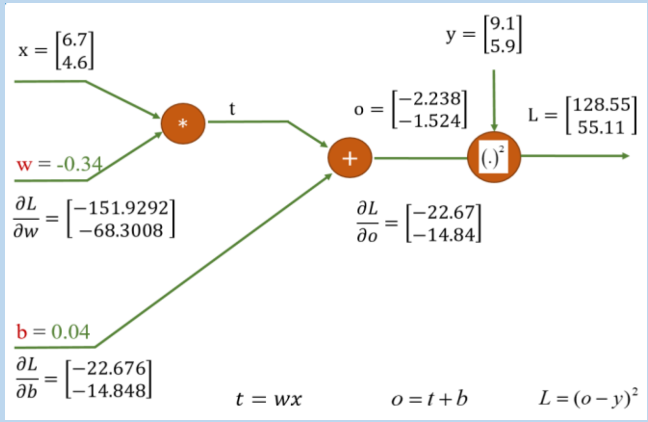
```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

Summary

Linear regression



Computational graph


$x = \begin{bmatrix}6.7\\4.6\end{bmatrix}$
$y = \begin{bmatrix}9.1\\5.9\end{bmatrix}$
$w = -0.34$
$\frac{\partial L}{\partial w} = \begin{bmatrix}-151.9292\\-68.3008\end{bmatrix}$
*
t
+
$o = \begin{bmatrix}-2.238\\-1.524\end{bmatrix}$
$(.)^2$
$L = \begin{bmatrix}128.55\\55.11\end{bmatrix}$
$\frac{\partial L}{\partial o} = \begin{bmatrix}-22.67\\-14.84\end{bmatrix}$
$b = 0.04$
$\frac{\partial L}{\partial b} = \begin{bmatrix}-22.676\\-14.848\end{bmatrix}$
$t = wx$
$o = t + b$
$L = (o - y)^2$

Formulae cheat sheet

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = \boldsymbol{\theta}^T \boldsymbol{x}$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

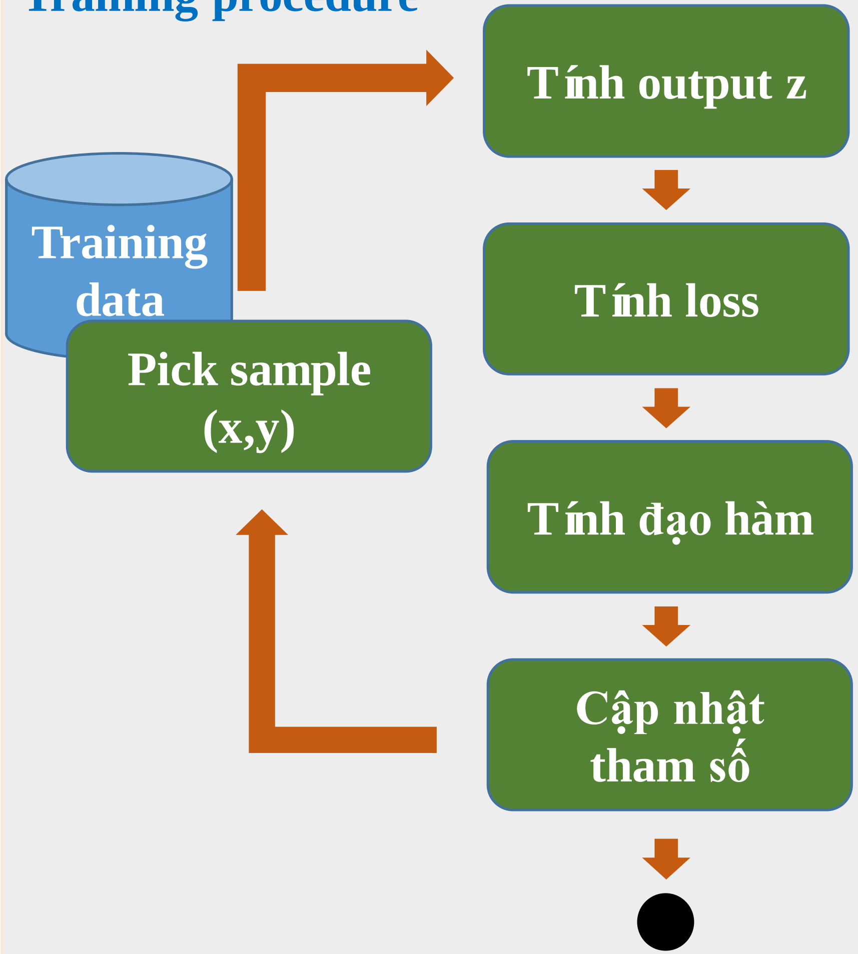
$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o - y)$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Training procedure



Thank you!