
Linear regression

— Tuan Nguyen - AI4E —

Problem

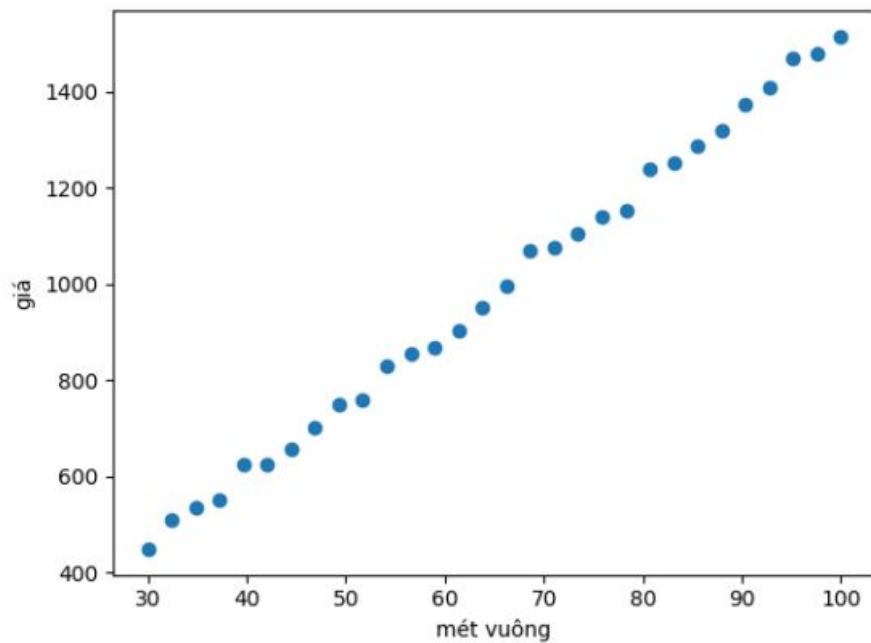


HOUSE PRICE PREDICTION

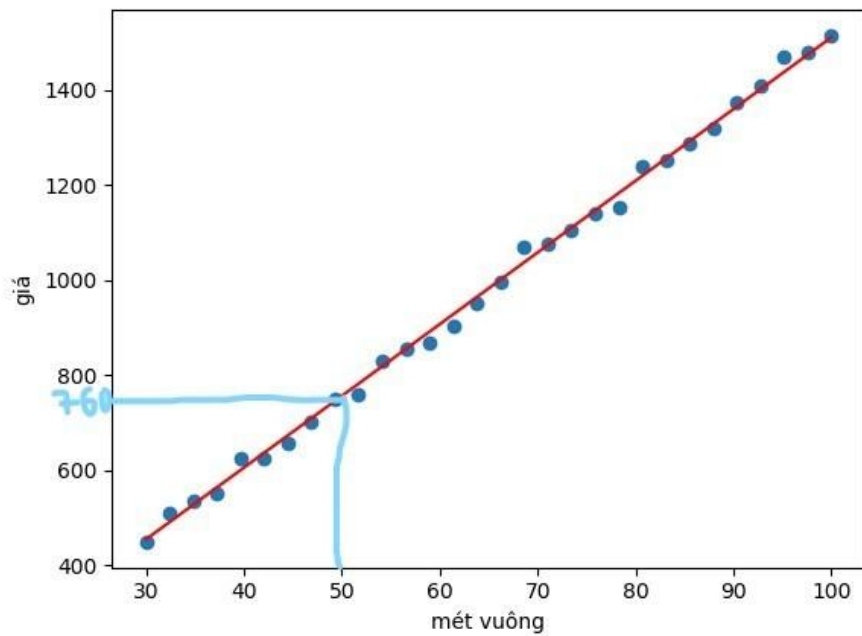
Dataset

Diện tích(m2)	Giá bán (triệu VNĐ)
30	448.524
32.4138	509.248
34.8276	535.104
37.2414	551.432
39.6552	623.418

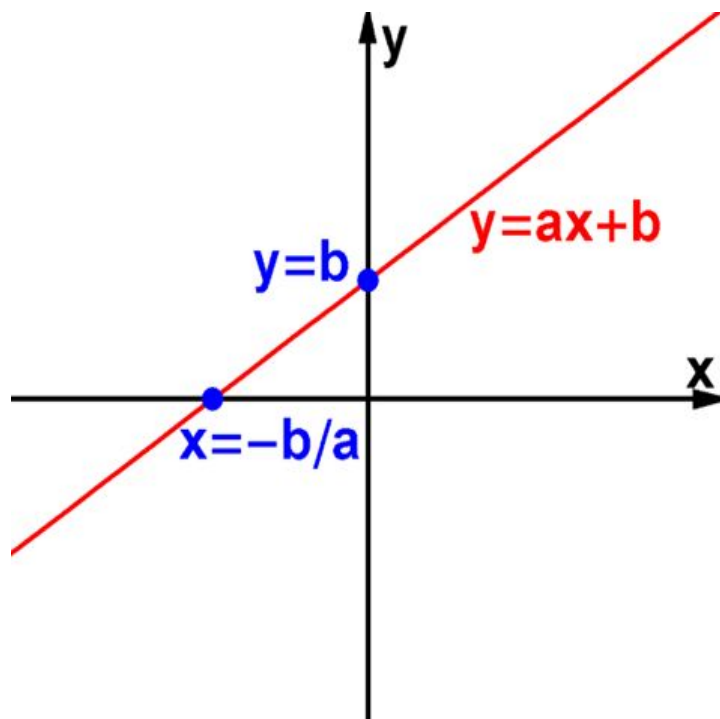
Visualization



How?

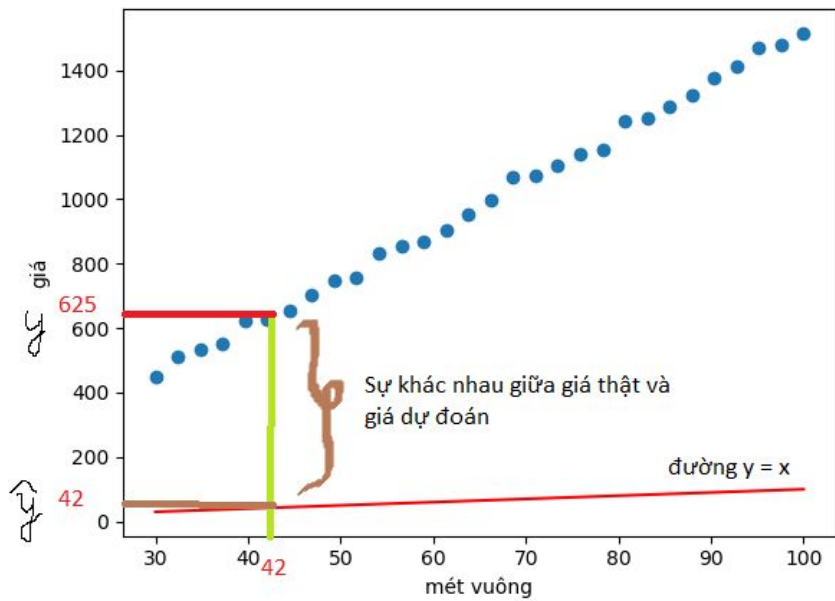


Line



$$y = w_1 * x + w_0$$

Loss function



Symbol

Diện tích(m2)		Giá bán (triệu VNĐ)	
x1	30	448.524	y1
x2	32.4138	509.248	y2
x3	34.8276	535.104	y3
...	37.2414	551.432	...
	39.6552	623.418	
	

Loss function

$$J = \frac{1}{2} * \frac{1}{N} * \left(\sum_{i=1}^N (\hat{y}_i - y_i)^2 \right)$$

$$\hat{y}_i = w_1 * x_i + w_0$$

- J không âm
- J càng nhỏ thì đường thẳng càng gần điểm dữ liệu.
- Nếu $J = 0$ thì đường thẳng đi qua tất các điểm dữ liệu.

Summary

1. Define the problem
2. Visualize the data
3. Choose the model
4. Loss function

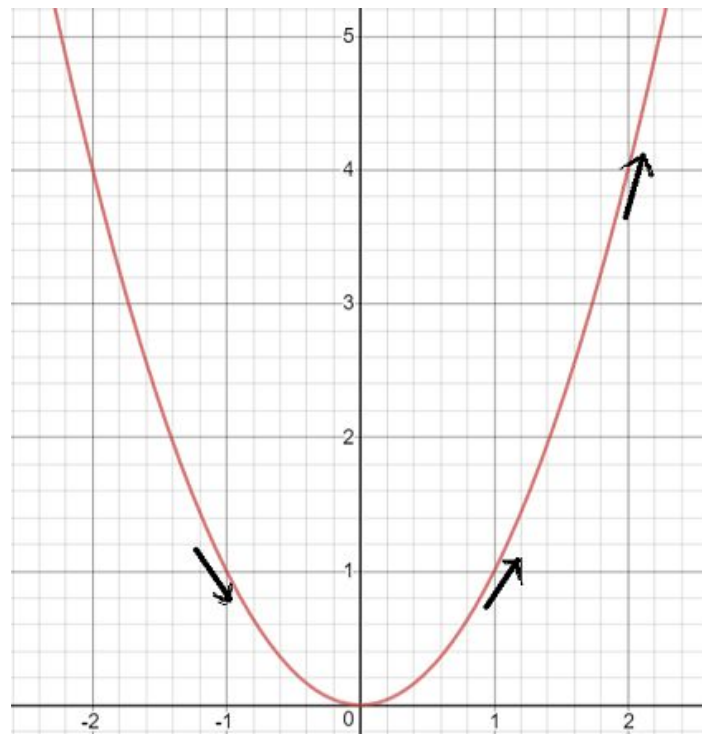
Solve the problem -> Find best parameter for the model -> Optimize loss function

How?

Gradient

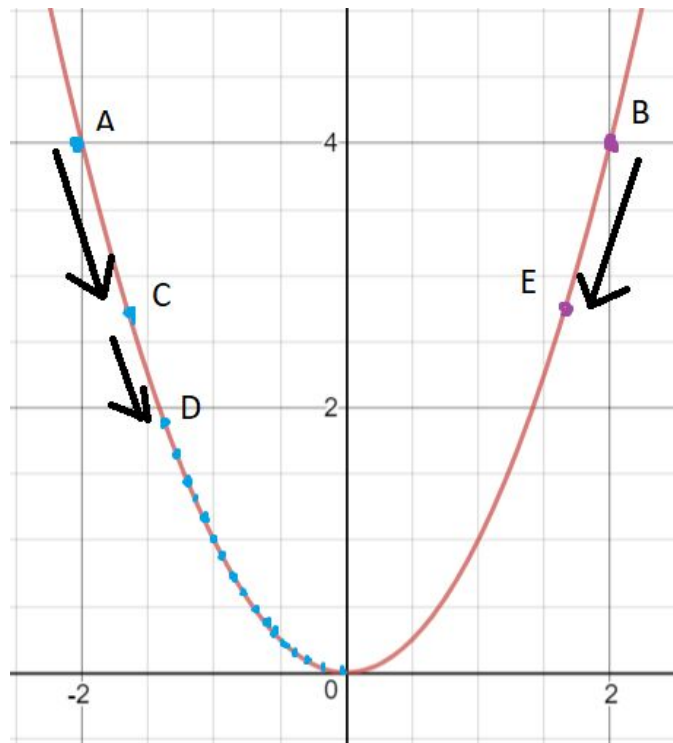
$$f(x) = x^2 \text{ là } f'(x) = \frac{df(x)}{dx} = 2 * x$$

- $f'(1) = 2 * 1 < f'(2) = 2 * 2 \Rightarrow$ đồ thị gần điểm $x = 2$ dốc hơn đồ thị gần điểm $x = 1 \Rightarrow$ trị tuyệt đối của đạo hàm tại một điểm càng lớn thì gần điểm đấy càng dốc.
- $f'(-1) = 2 * (-1) = -2 < 0 \Rightarrow$ đồ thị đang giảm hay khi tăng x thì y sẽ giảm; ngược lại đạo hàm tại điểm nào đó mà dương thì đồ thị quanh điểm đấy đang tăng.

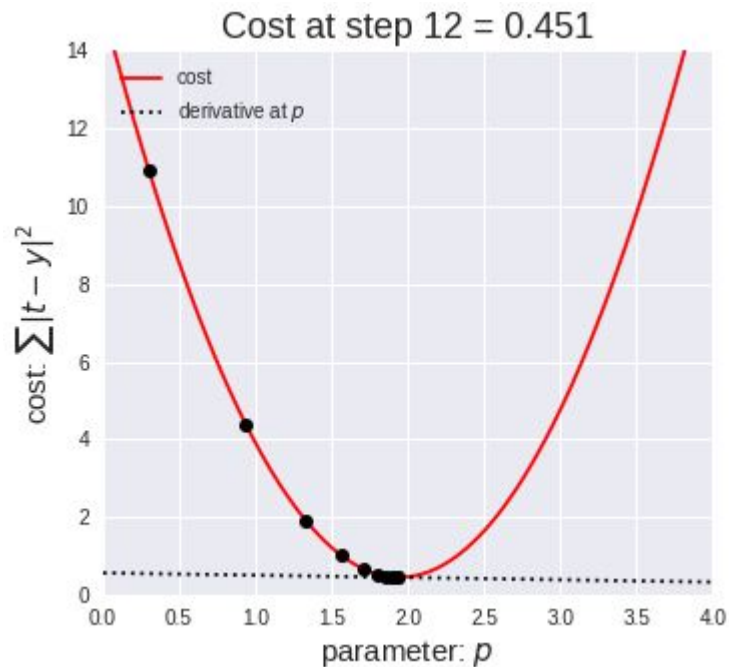


Gradient descent

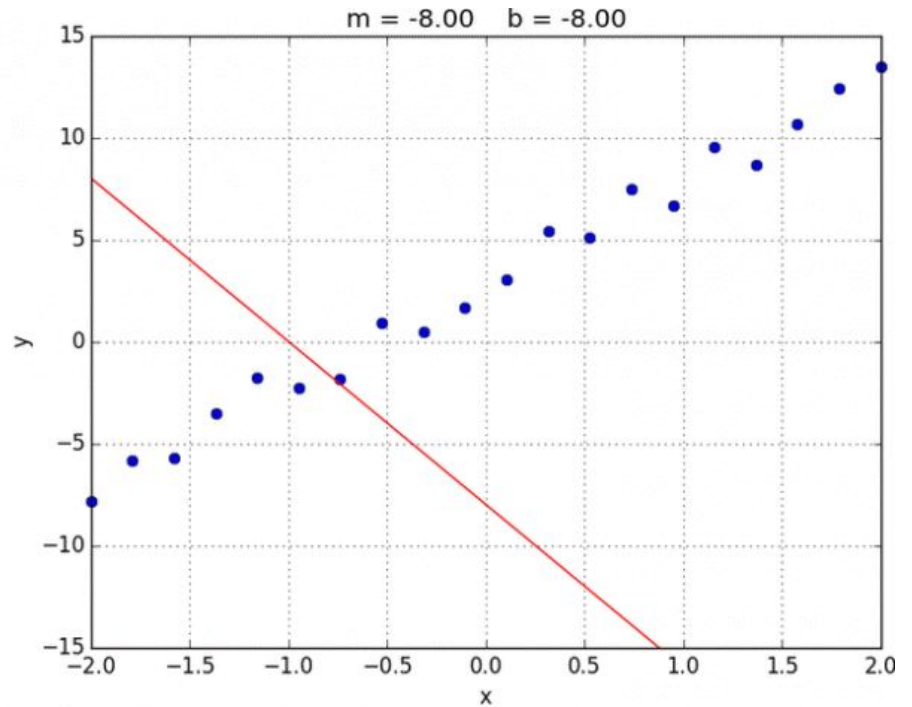
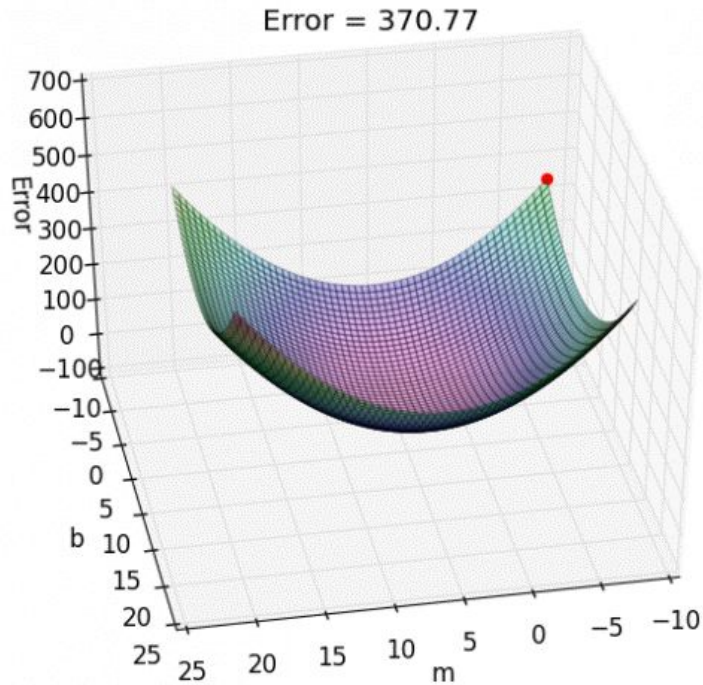
- Bước 1: Khởi tạo giá trị x tùy ý
- Bước 2: Gán $x = x - \text{learning_rate} * f'(x)$ (learning_rate là hằng số không âm ví dụ $\text{learning_rate} = 0.001$)
- Bước 3: Tính lại $f(x)$:
 - Nếu $f(x)$ đủ nhỏ thì dừng lại.
 - Ngược lại tiếp tục bước 2.



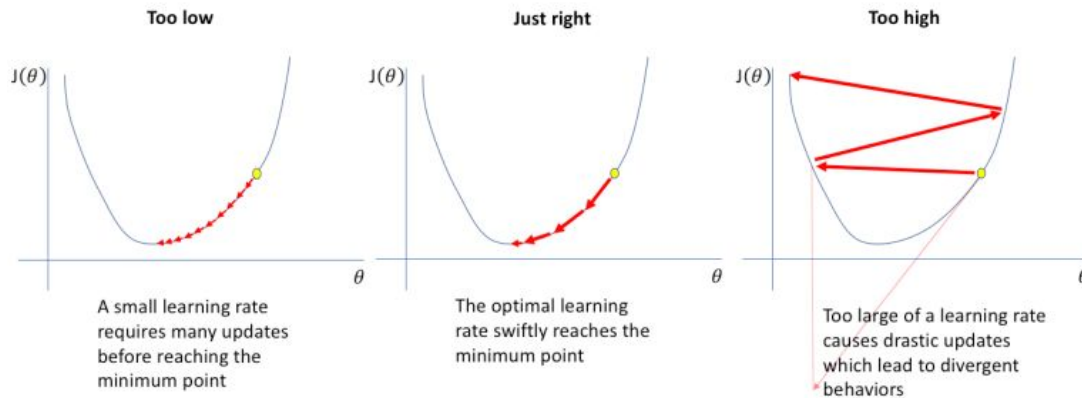
Gradient descent visualization



Gradient descent visualization

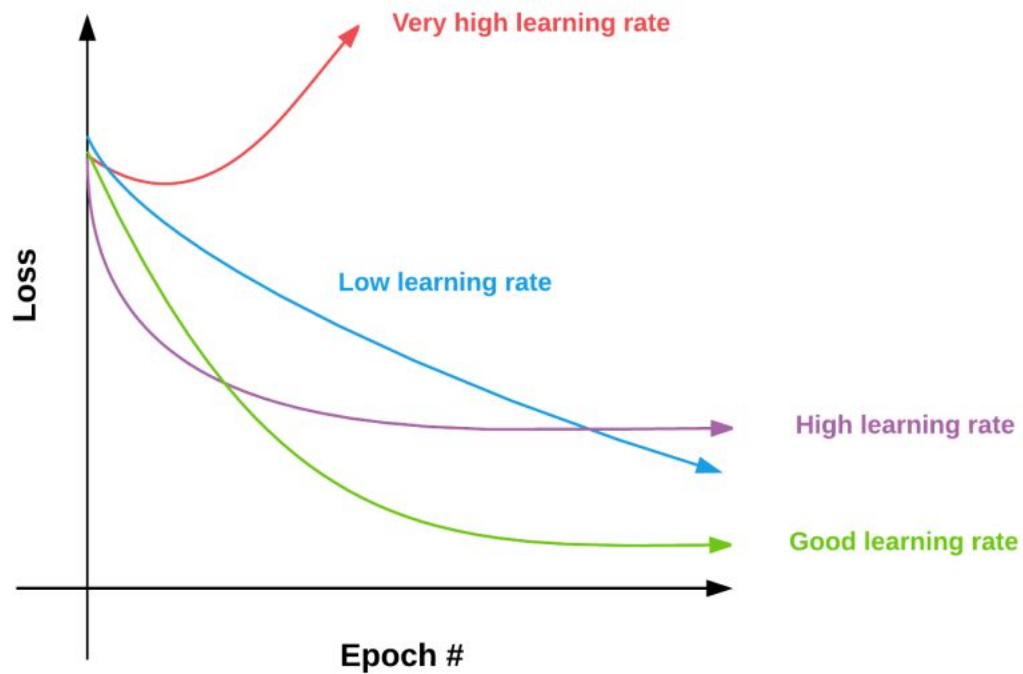


Check learning rate



- Nếu `learning_rate` nhỏ: mỗi lần hàm số giảm rất ít nên cần rất nhiều lần thực hiện bước 2 để hàm số đạt giá trị nhỏ nhất
- Nếu `learning_rate` hợp lý: sau một số lần lặp bước 2 vừa phải thì hàm sẽ đạt giá trị đủ nhỏ.
- Nếu `learning_rate` quá lớn: sẽ gây hiện tượng overshoot và không bao giờ đạt được giá trị nhỏ nhất của hàm.

Check loss function



Gradient review

$$(x^a)' = a * x^{a-1}$$

$$(a^x)' = a^x * \ln(a)$$

$$(e^x)' = e^x$$

$$(\ln(x))' = \frac{1}{x}$$

$$(\sin(x))' = \cos(x)$$

$$(\cos(x))' = -\sin(x)$$

Calculate gradient

$$J(w_0, w_1) = \frac{1}{2} * \left(\sum_{i=1}^N (\hat{y}_i - y_i)^2 \right) = \frac{1}{2} * \left(\sum_{i=1}^N (w_0 + w_1 * x_i - y_i)^2 \right)$$

$$\frac{dJ}{dw_0} = \sum_{i=1}^N (w_0 + w_1 * x_i - y_i)$$

$$\frac{dJ}{dw_1} = \sum_{i=1}^N x_i * (w_0 + w_1 * x_i - y_i)$$

Vector presentation

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}, W = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

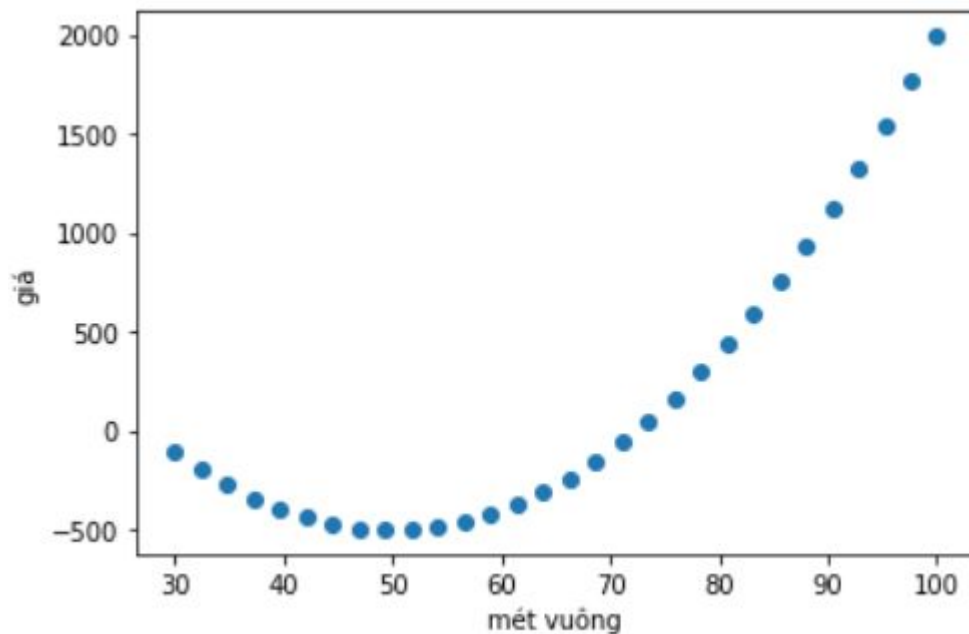
$$\hat{Y} = X * W = \begin{bmatrix} w_0 + w_1 * x_1 \\ w_0 + w_1 * x_2 \\ \dots \\ w_0 + w_1 * x_n \end{bmatrix}$$

Gradient

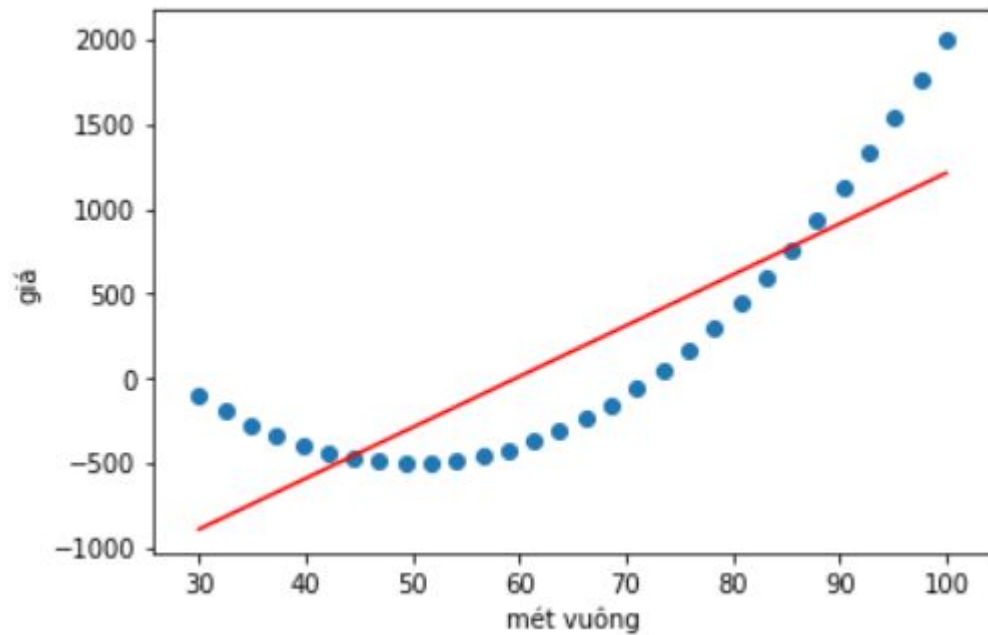
$$X[:, 1] = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \text{sum}(X[:, 1]) = x_1 + x_2 + \dots + x_n$$

$$\frac{dJ}{dw_0} = \text{sum}(\hat{Y} - Y), \frac{dJ}{dw_1} = \text{sum}(X[:, 1] \otimes (\hat{Y} - Y))$$

Non-linear



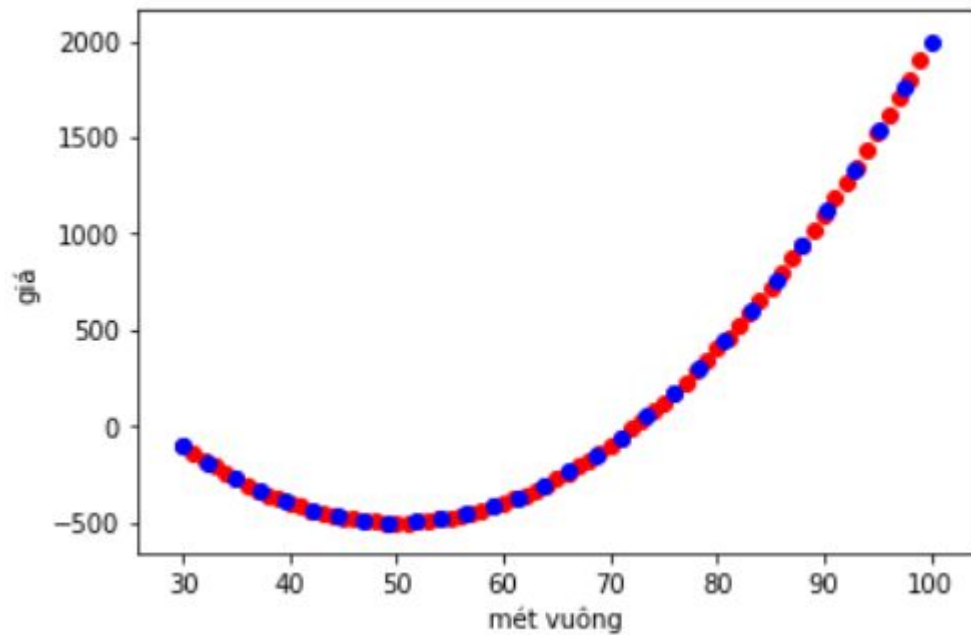
How to fit?



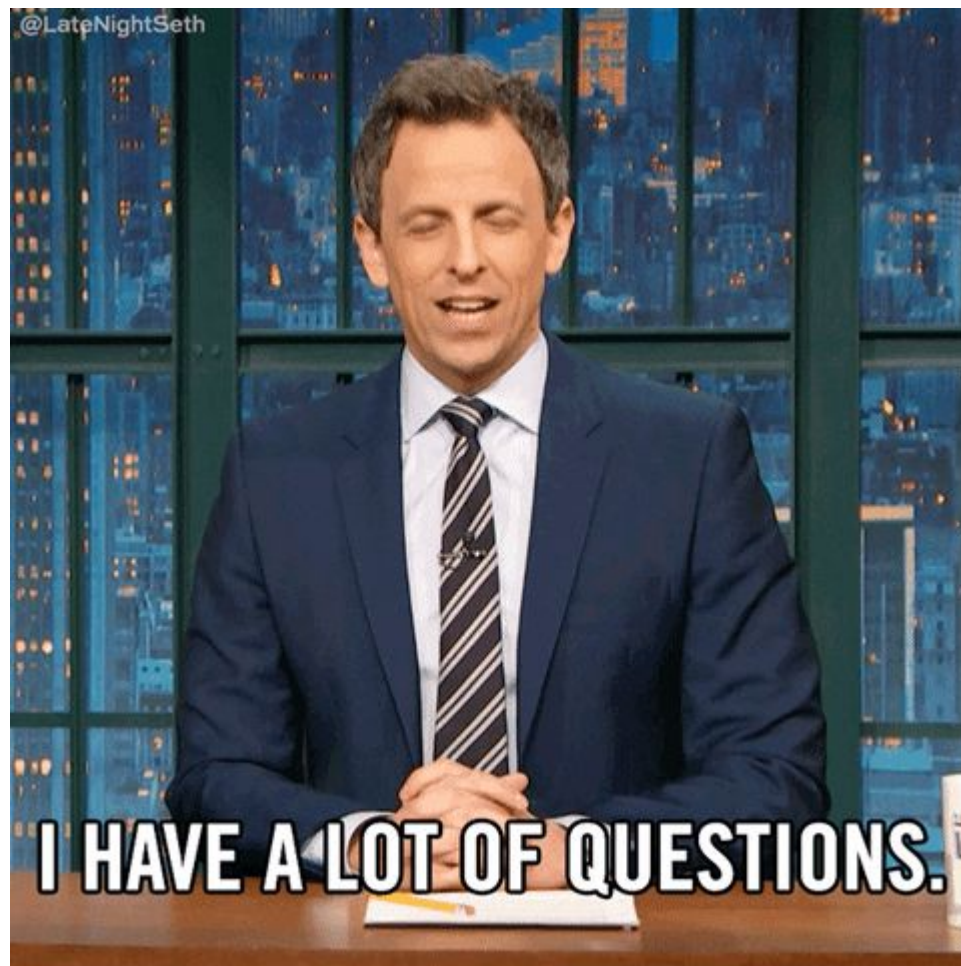
Matrix representation

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} \Rightarrow X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots & \dots & \dots \\ 1 & x_n & x_n^2 \end{bmatrix}$$
$$W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Parabol



Q&A





Thank you.