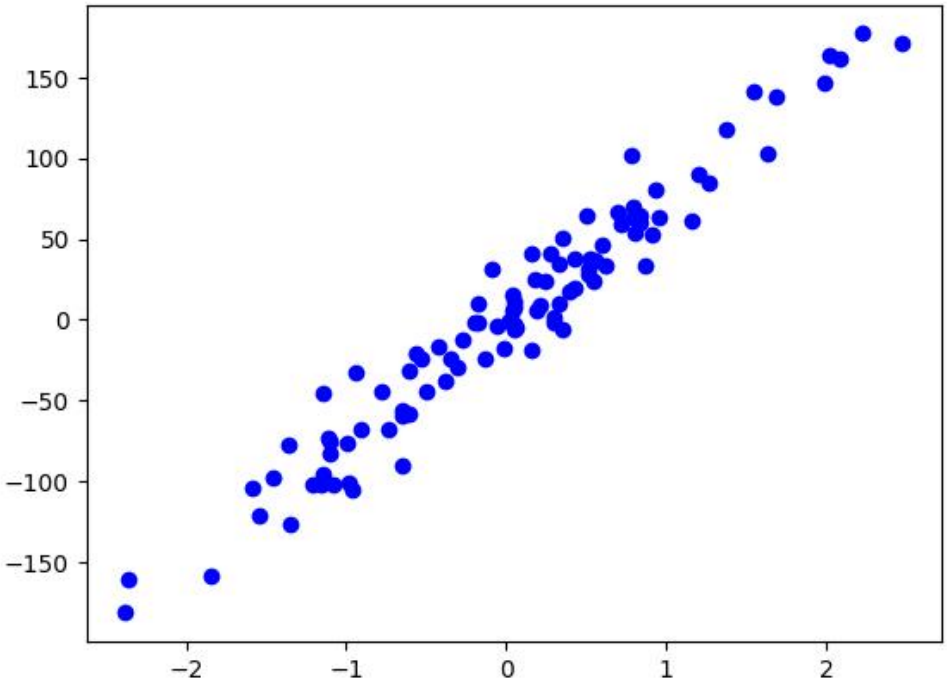
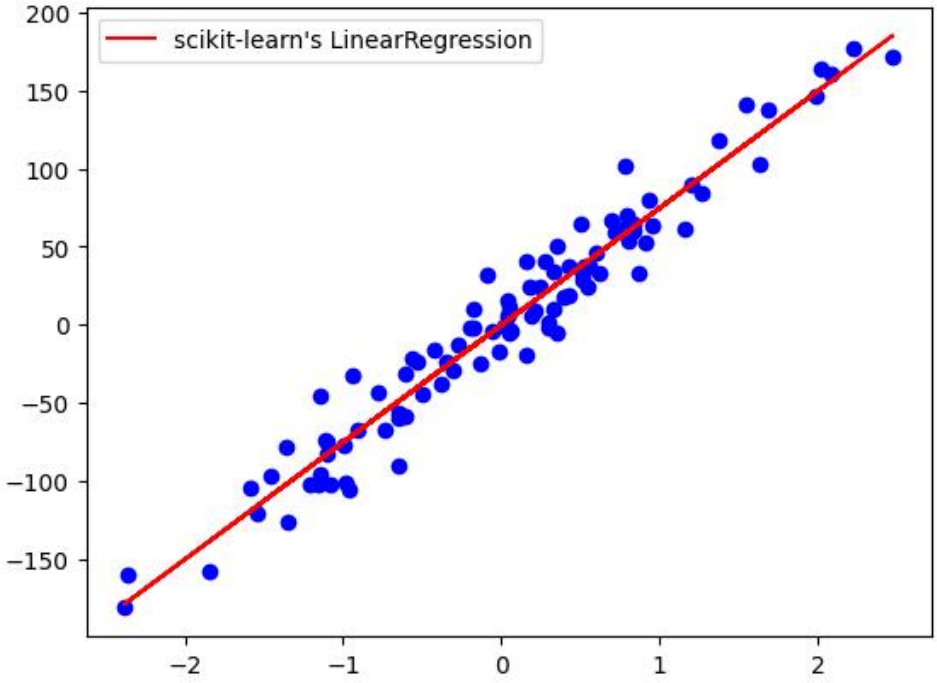
Linear Regression (Hồi quy tuyến tính)



Ở hình trên là một tập dữ liệu với mẫu tuyến tính, và mình có nhiệm vụ rằng là mình phải sử dụng thuật toán hồi quy tuyến tính để vẽ một đường thẳng tuyến tính phù hợp với cách mà các điểm dữ liệu trình bày trong tập dữ liệu ấy.



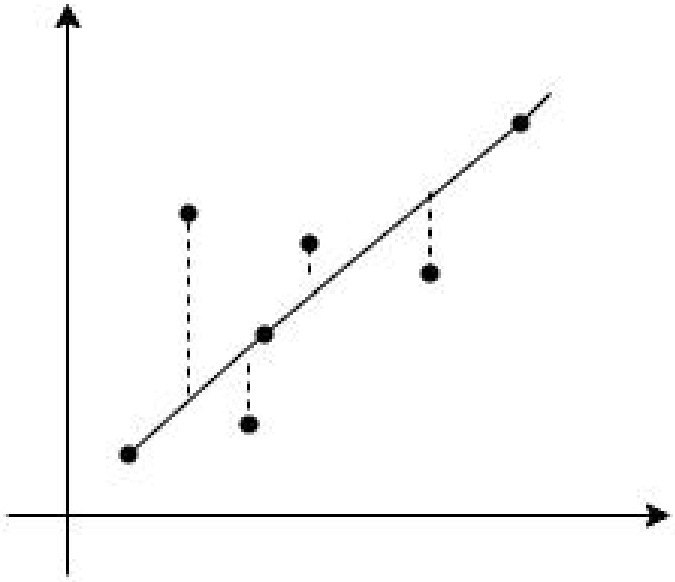
Mình có công thức rằng:

$$\hat{y} = w.x + b$$

Trong đó:

- $\hat{y}$: kết quả dự đoán/đầu ra
- w : trọng số (weight)
- x : đầu vào
- b : bias

Với đầu ra có được, mình thường thì sẽ đi tính sai số của dữ liệu bằng cách tính khoảng cách từ các điểm dữ liệu cho đến cái đường thẳng tuyến tính được vẽ.



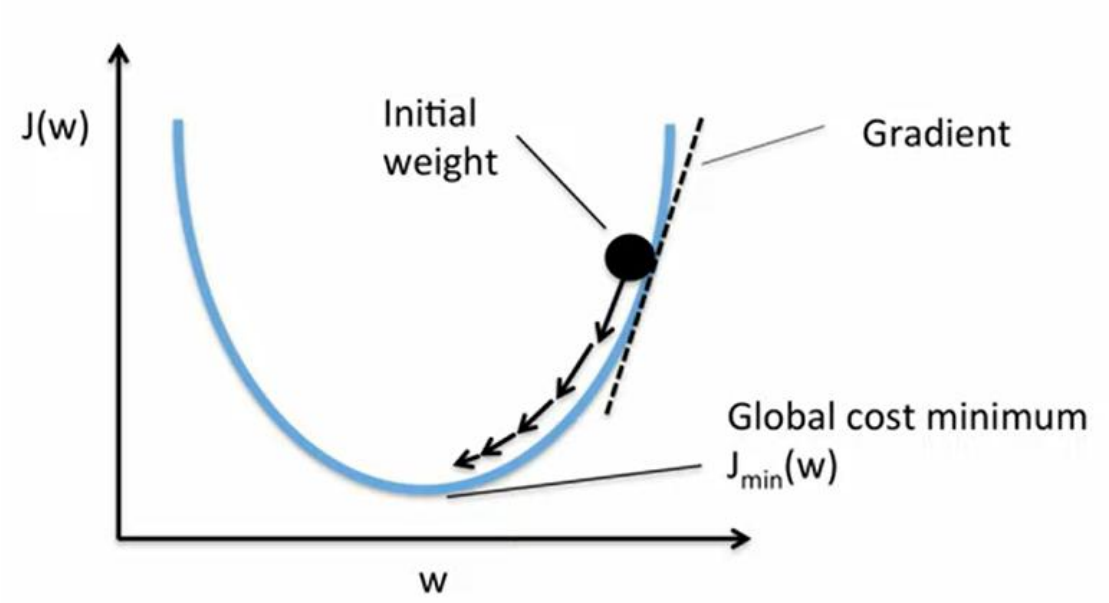
Các sai số là khoảng cách (đường kẻ vết dứt) từ các điểm dữ liệu (các chấm đen) đến đường thẳng tuyển tính. Mình có công thức tính sai số trunh bình (hàm mất mát), với n số lượng mẫu/điểm dữ liệu như sau:

$$\text{MSE} = \text{J}(\text{w}, \text{b}) = 1/\text{n} * \sum_{\text{i}=1}^{\text{n}} (\text{y}_\text{i} - (\text{wx}_\text{i} + \text{b}))^2$$

Với công thức trên, mình có thể tính được sai số bình phương trung bình (mean squared error), và mình sẽ muốn rằng nó sẽ cho ra kết quả sai số tối thiểu nhất cho thuật toán, tương đương với việc tìm ra 2 thông số tốt nhất trong quá trình huấn luyện: trọng số (w) và bias (b). Nhưng để thật sự tối thiểu nó, mình tính đạo hàm, hoặc độ dốc (gradient), của sai số bình phương trunh bình, rồi mình có thể tối ưu hóa trọng số và bias:

$$J^{'}(w,b) = \begin{bmatrix} \frac{dJ}{dw} \\ \frac{dJ}{db} \end{bmatrix} = \begin{bmatrix} \frac{-2}{n}\sum_{i=1}^{n} x_i(y_i - (wx_i + b)) \\ \frac{-2}{n}\sum_{i=1}^{n} (y_i - (wx_i + b)) \end{bmatrix}$$

Để làm được kỹ thuật tối ưu hóa trên, mình sử dụng kỹ thuật Gradient Descent. Gradient Descent là một thuật toán tối ưu được sử dụng để giảm thiểu sai số trong học máy bằng cách tìm trọng số tối ưu cho mô hình.

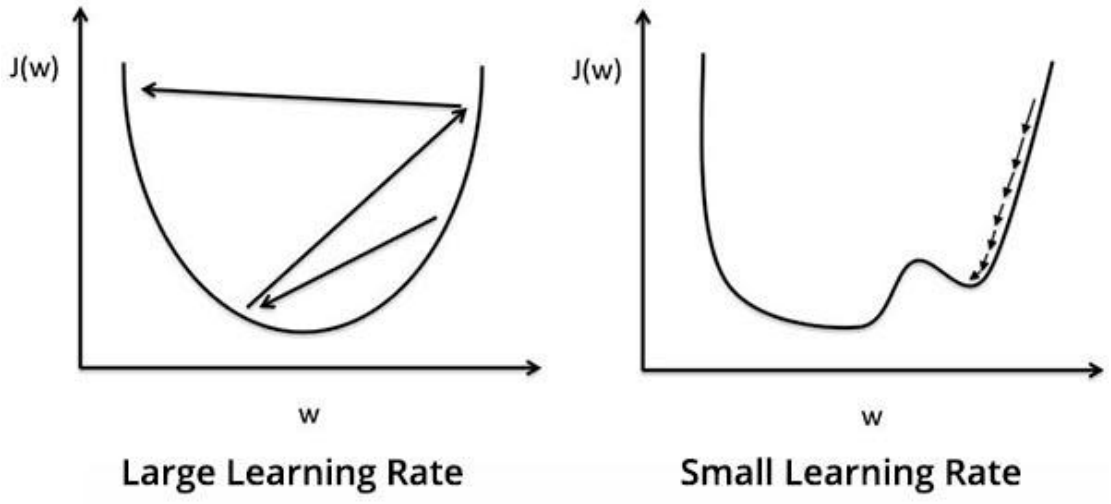


Từ sơ đồ trên, với một giá trị thông số được cho w và sai số đã được tính J(w), mình tính (tại một điểm mà mình có giá trị thông số Initial weight) hướng để đi của hàm mất mát để giảm thiểu MSE, hoặc nói chung là lỗi của cả mô hình. Sau khi mình, tính được đạo hàm của hàm mất mát, mình sẽ cập nhật trọng số và bias với một tỷ lệ học (learning rate) nhất định.

$$\mathrm{w} = \mathrm{w} - \alpha * \frac{dJ}{dw}$$

$$\mathrm{b} = \mathrm{b} - \alpha * \frac{dJ}{db}$$

Tỷ lệ học α cho mình biết được nó đi xuống dốc nhanh hay chậm theo hướng thuật toán Gradient Descent muốn.



Nguồn ảnh: https://saugatbhattarai.com.np/what-is-gradient-descent-in-machine-learning/

Với tỷ lệ học lớn, nó sẽ nhảy loạn xạ thấp nơi đến mức mà mình sẽ không bao giờ đạt được sai số tối thiểu nhất. Còn đối với tỷ lệ học nhỏ, nó có thể gây ra vấn đề về việc đạt sai số nhỏ nhất bởi vì mình đang tiếp cận nó một cách chậm rãi.

Sơ đồ biểu diễn quá trình huấn luyện của thuật toán: