

Slide Report Week 1

Le Viet Hung

AI Intern

Content

Gradient Descent

Batch Gradient Descent

Mini-batch Gradient Descent

Stochastic Gradient Descent,

Overfitting and Underfitting

Gradient Descent

Giới thiệu: Thuật toán Gradient Descent được sử dụng phổ biến trong các bài toán tối ưu, chẳng hạn như machine learning, thông thường việc giải một phương trình đạo hàm để tìm cực tiểu tối ưu là khó đặc biệt là những phương trình phức tạp. Gradient Descent ra đời để giải quyết các bài toán tìm cực tiểu tối ưu một cách dễ dàng hơn.

Gradient Descent

Gradient Descent là một thuật toán tối ưu được sử dụng rộng rãi trong machine learning. Thuật toán này được sử dụng để tìm kiếm điểm tối ưu của một hàm mất mát (loss function) thông qua việc điều chỉnh các tham số của mô hình dự đoán.

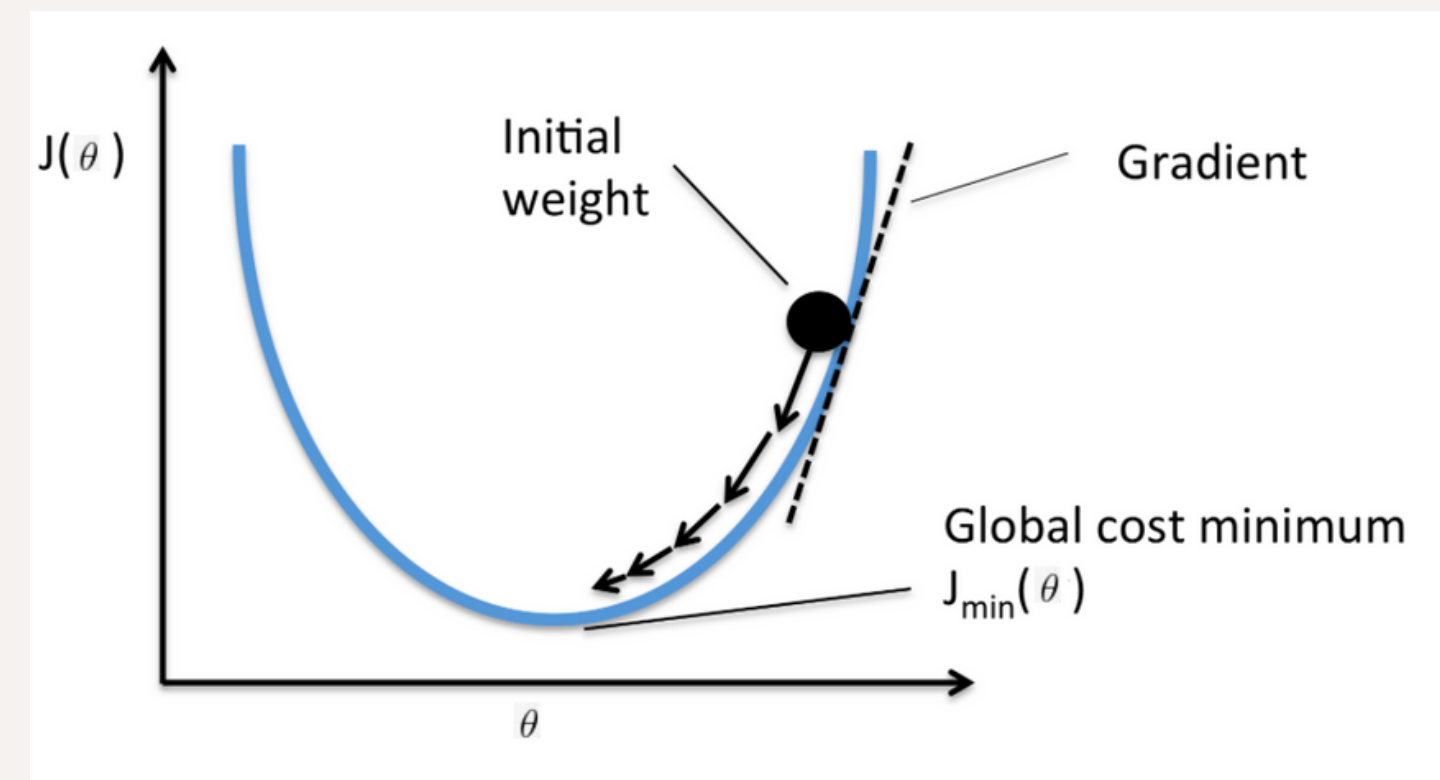
Công thức toán học của thuật toán Gradient Descent được biểu diễn như sau:

$$\theta = \theta - \alpha \nabla_{\theta} J(\theta; \mathbf{x}; \mathbf{y})$$

θ là tham số được cập nhật sau mỗi lần lặp

α là learning rate, hay còn gọi là tốc độ cập nhật

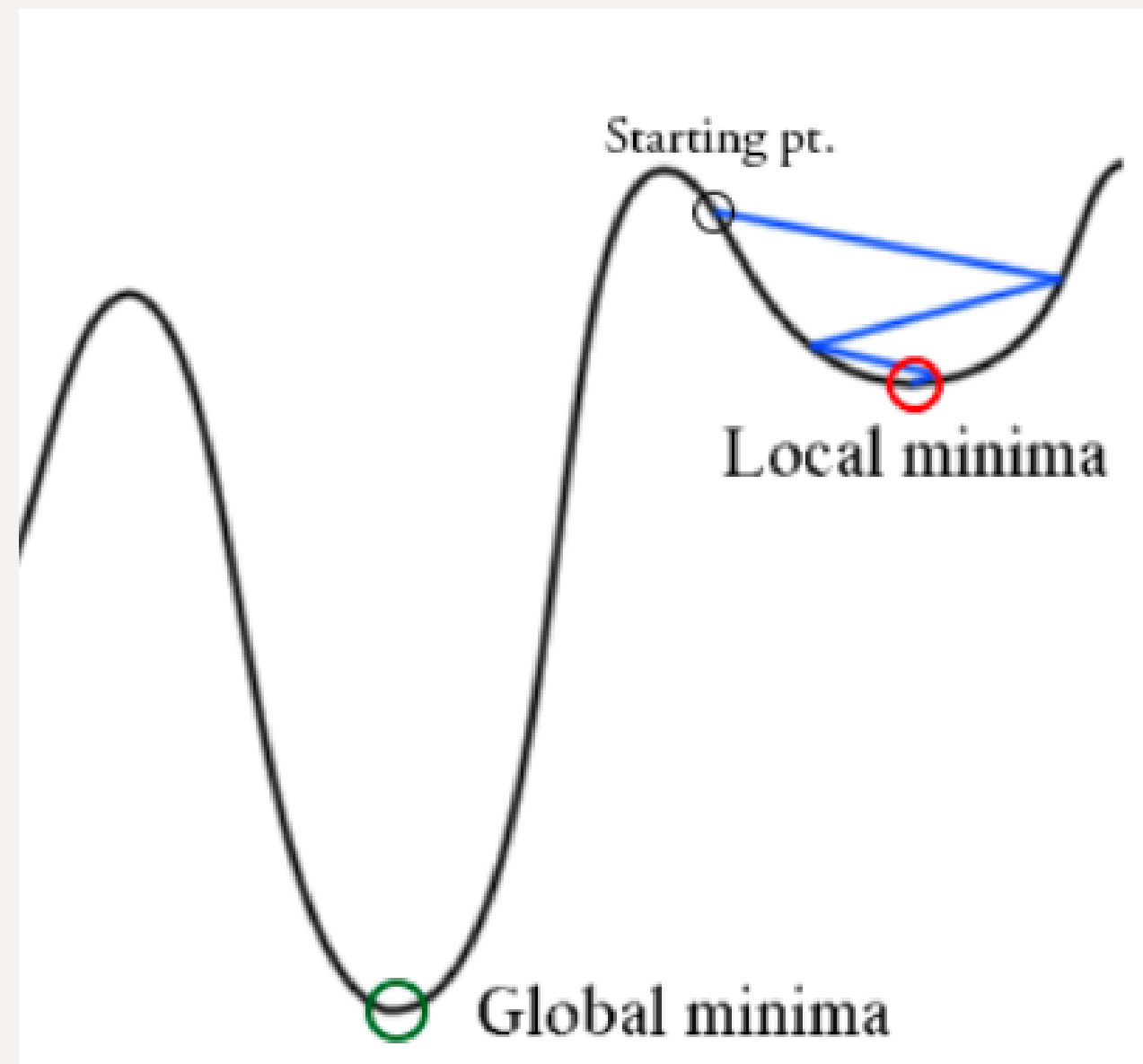
$\nabla_{\theta} J(\theta; \mathbf{x}; \mathbf{y})$ là đạo hàm của hàm loss J theo tham số theta



Gradient Descent

Vấn đề thường gặp trong Gradient Descent: Cực tiểu cục bộ (local minimum)

Vấn đề cực tiểu cục bộ là một thách thức trong quá trình tối ưu hóa bằng Gradient Descent. Khi áp dụng Gradient Descent để tìm kiếm điểm cực tiểu của hàm mất mát, có khả năng thuật toán sẽ dừng lại ở một điểm cực tiểu cục bộ thay vì đạt được điểm cực tiểu toàn cục mong muốn.



Gradient Descent

Gọi N là số lượng mẫu trong tập dữ liệu train
Ta có bảng so sánh sau:

Batch Gradient Descent	Mini-batch Gradient Descent	Stochastic Gradient Descent
Số sample train mỗi bước lặp : $m=N$ sample	Số sample train mỗi bước lặp : $1 < m < N$ sample	Số sample train mỗi bước lặp : $m = 1$ sample
Khối lượng tính toán mỗi lần lặp lớn	Khối lượng tính toán mỗi lần lặp lớn	Khối lượng tính toán mỗi lần lặp bé
Tốc độ thực thi nhanh	Tốc độ thực thi nhanh	Tốc độ thực thi chậm
Loss hội tụ chậm	Loss hội tụ chậm	Loss hội tụ nhanh

Gradient Descent

Minh họa:

epochs

{

iterations i in N/m

{

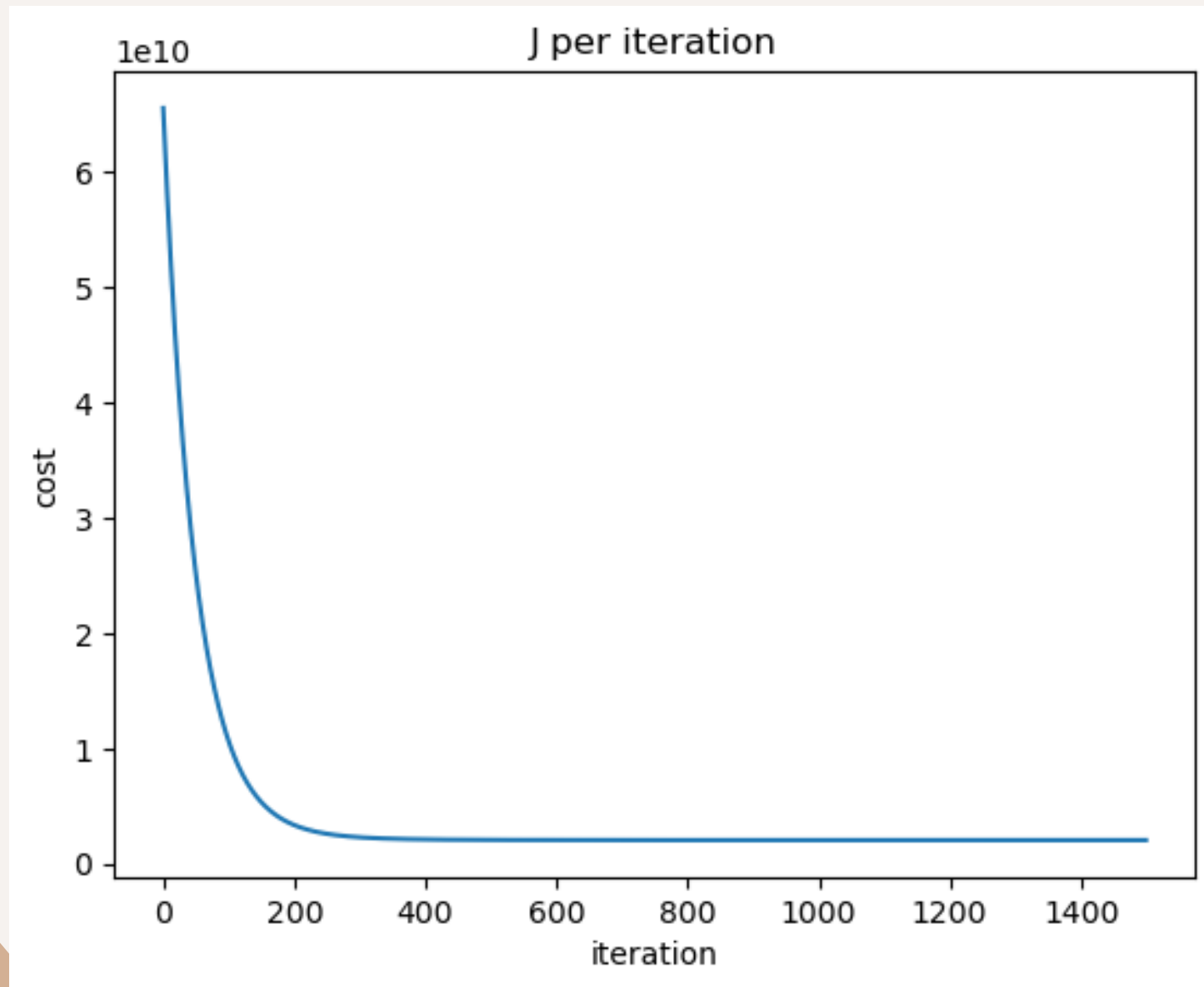
training m sample

}

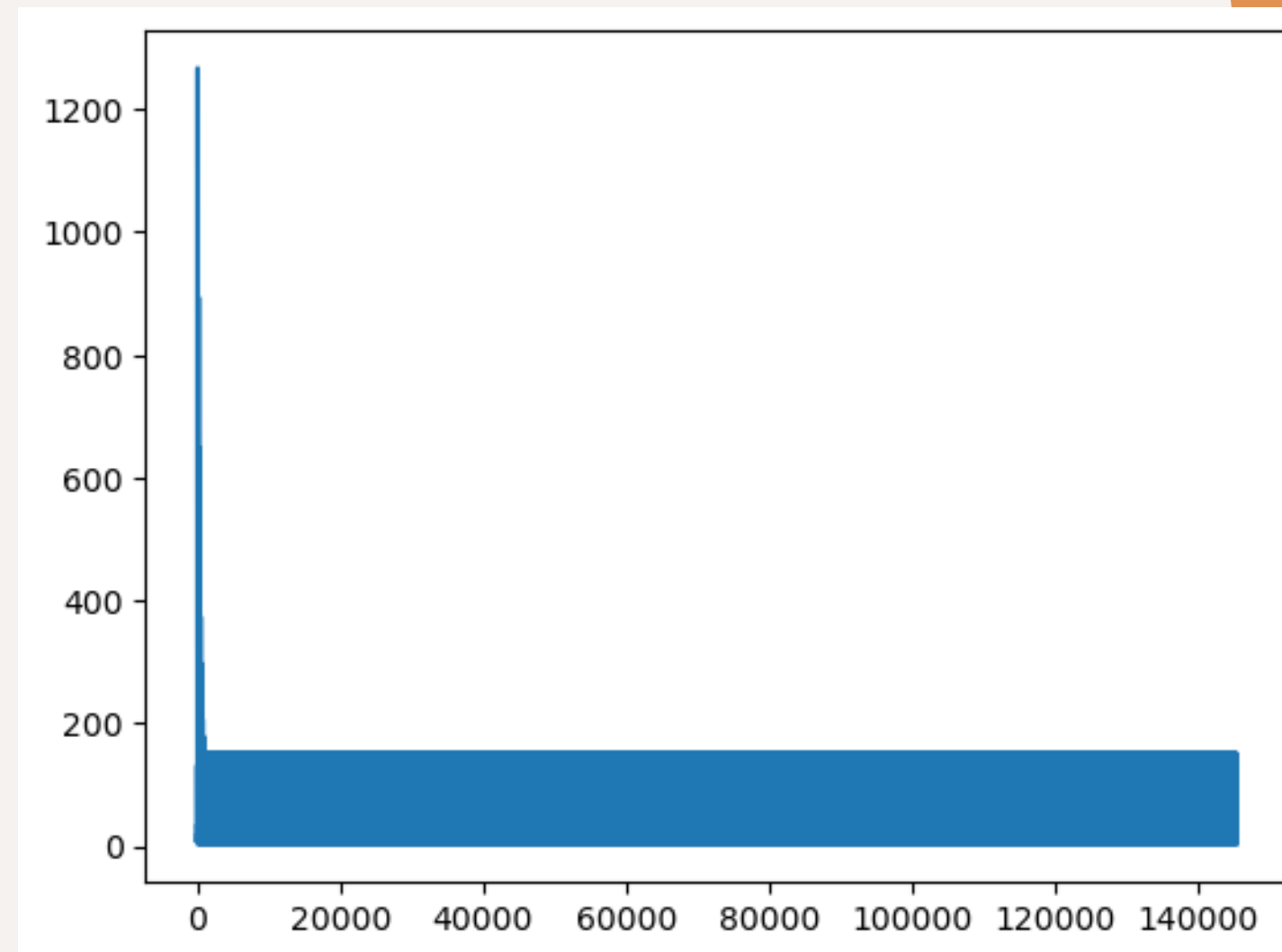
}

Gradient Descent

So sánh đồ thị Loss giữa Batch Gradient Descent và Stochastic Gradient Descent:



Batch Gradient Descent



Stochastic Gradient Descent

Overfitting and Underfitting

Overfitting : có nghĩa là mô hình có khả năng chỉ hoạt động tốt trên tập dữ liệu huấn luyện mà không tốt trên tập dữ liệu thực tế.

Underfitting : là tình trạng khi một mô hình không đủ mạnh để học được trên tập dữ liệu huấn luyện

Overfitting and Underfitting

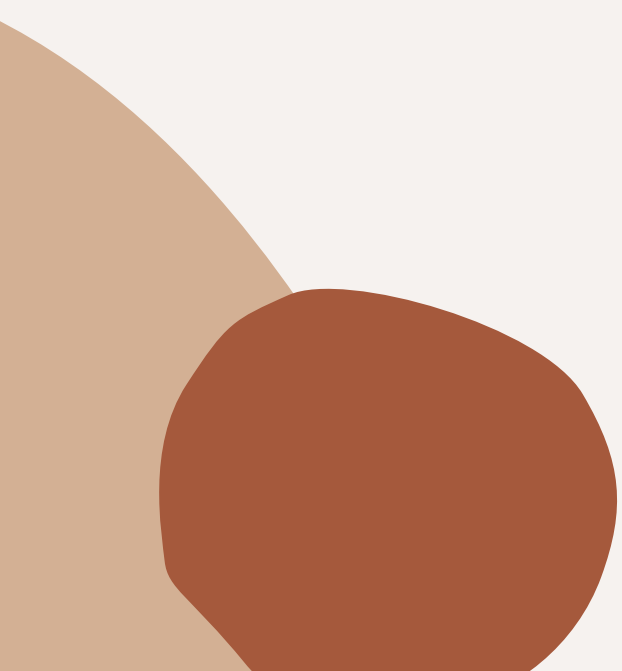
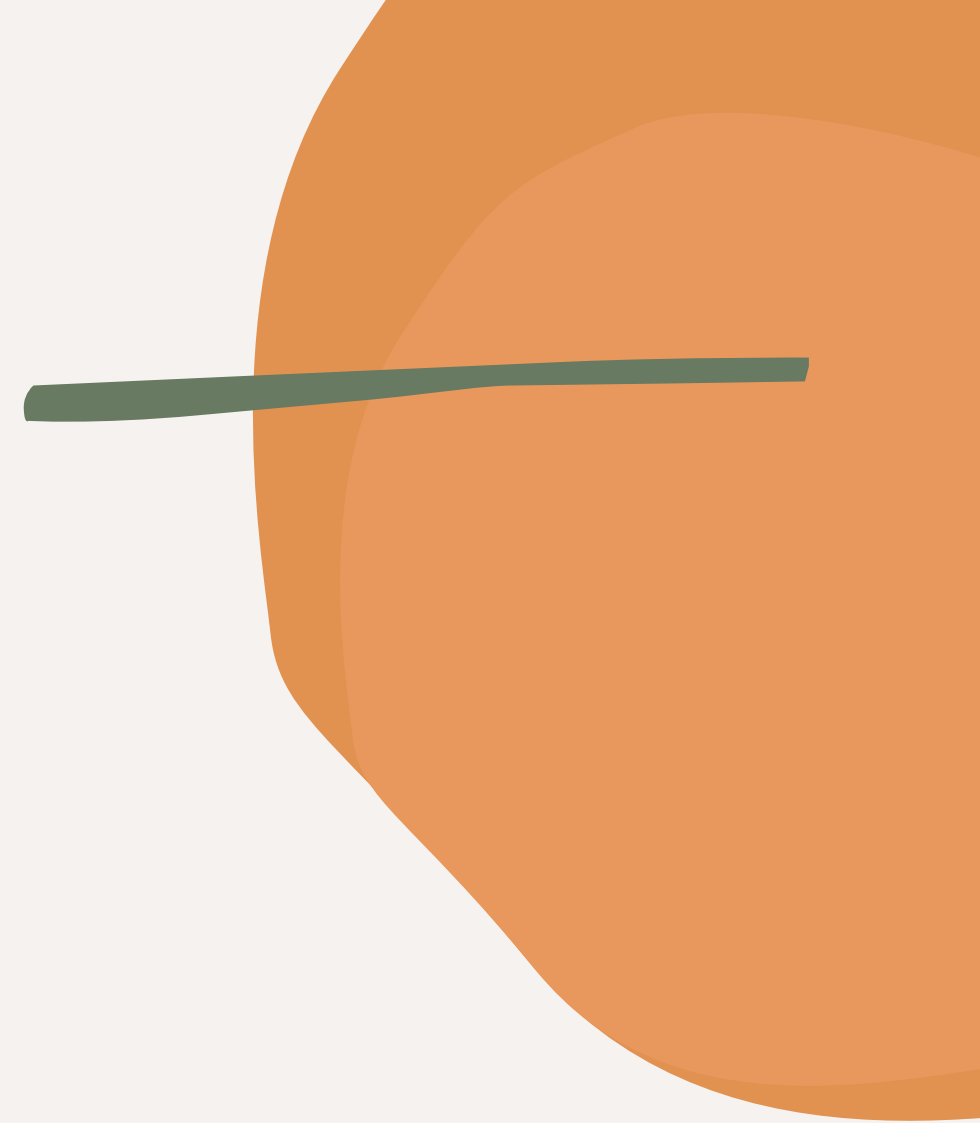
Nguyên nhân:

Overfitting:

- Mô hình quá phức tạp
- Dữ liệu quá khác so với dữ liệu thực tế

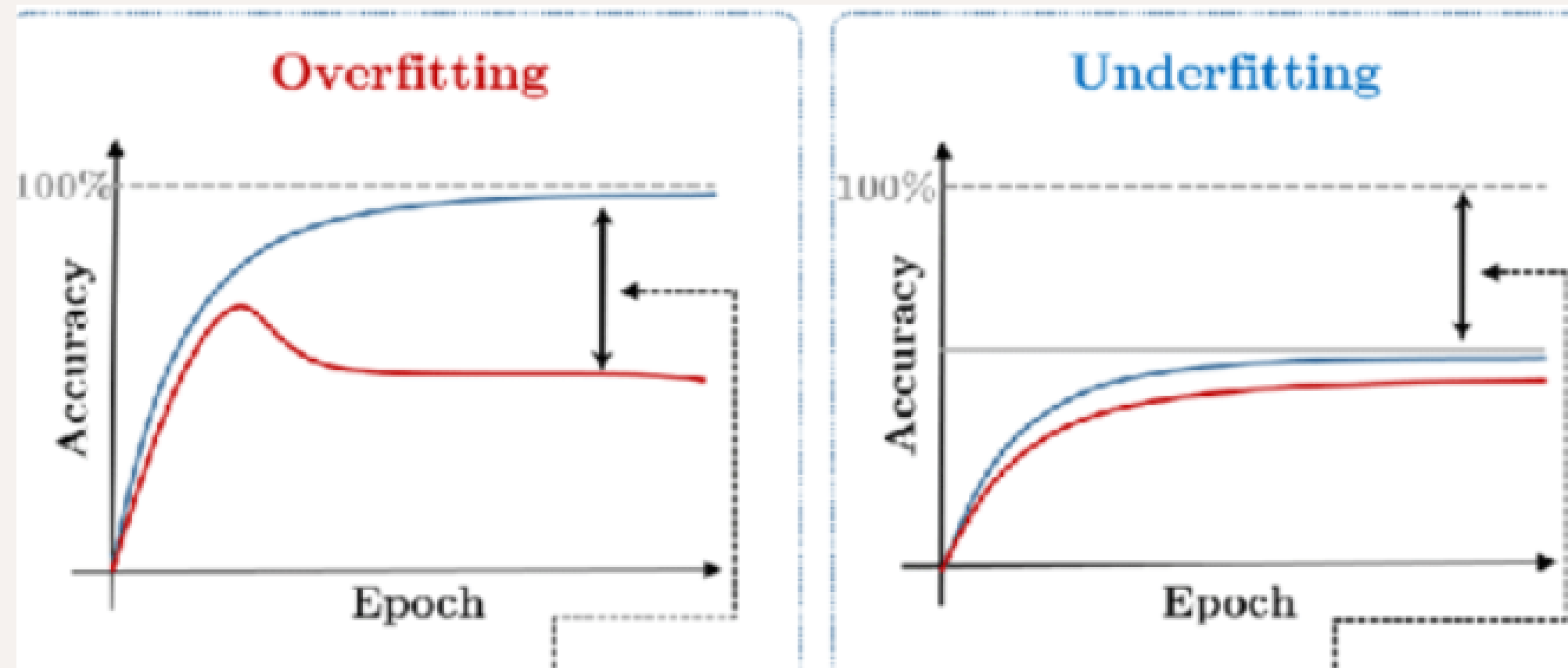
Underfitting:

- Mô hình quá đơn giản
- Dữ liệu quá nhỏ, đơn giản, không đủ phức tạp



Overfitting and Underfitting

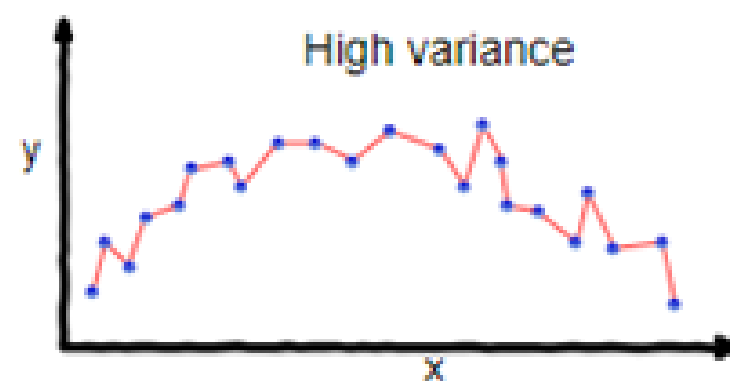
Minh họa Overfitting và Underfitting trên Learning curves:



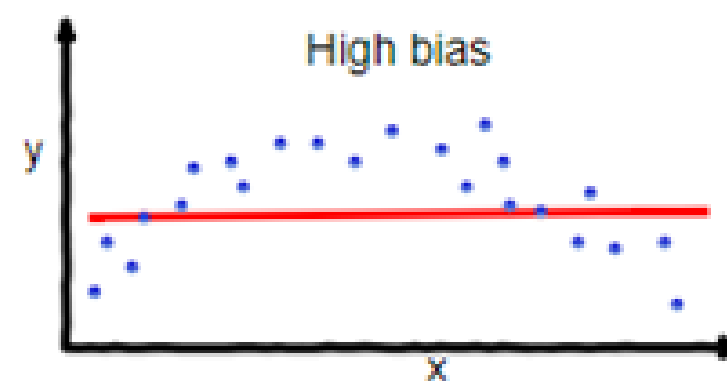
Ở đây, đường màu xanh là accuracy trên tập train và đường màu đỏ là accuracy trên tập test
Overfitting xảy ra khi accuracy trên tập training quá cao và cao hơn nhiều so với tập test
Underfitting xảy ra khi accuracy trên cả tập train và tập test đều thấp

Overfitting and Underfitting

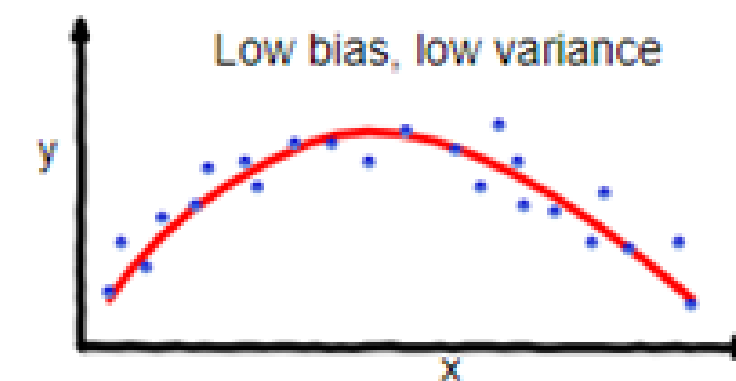
Minh họa Overfitting và Underfitting trên trong bài toán hồi quy:



overfitting



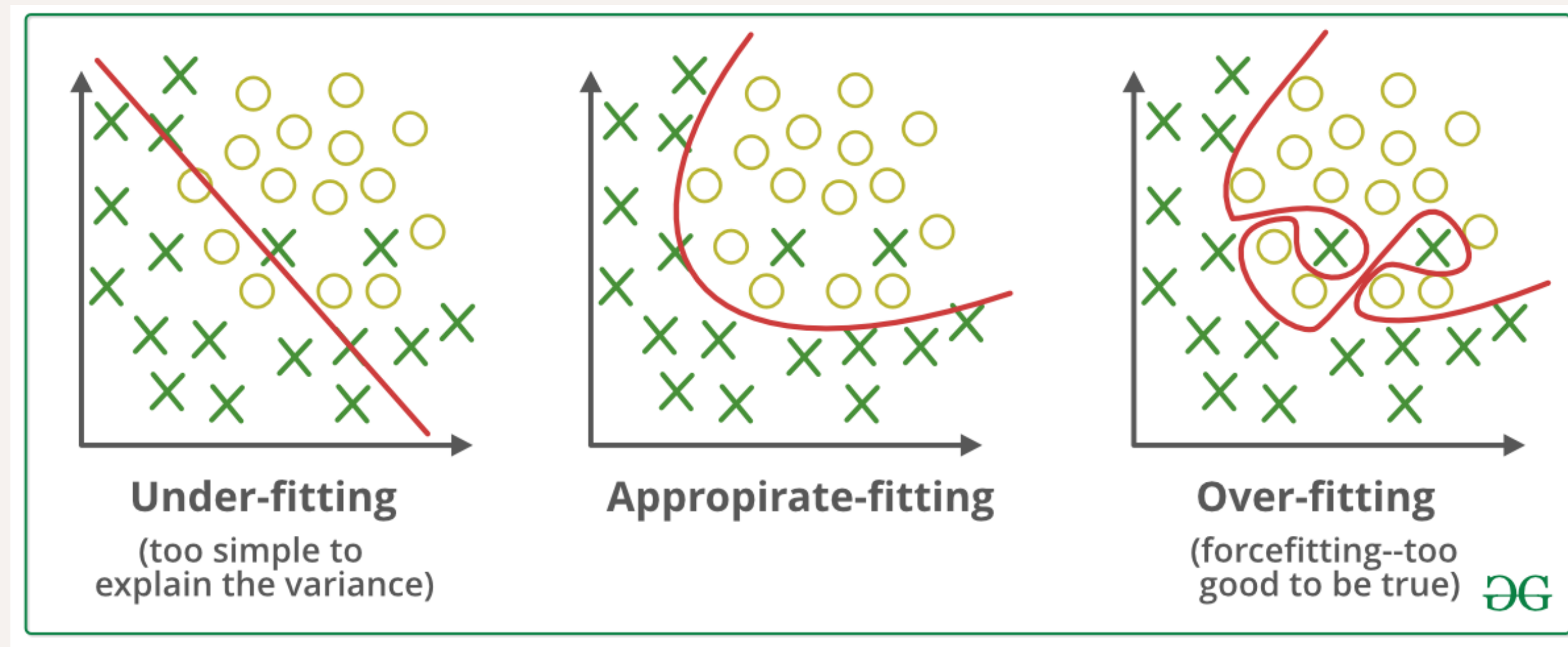
underfitting



Good balance

Overfitting and Underfitting

Minh họa Overfitting và Underfitting trên trong bài toán phân loại:



Overfitting and Underfitting

Phương pháp làm giảm Overfitting và Underfitting :

Overfitting:

- Regularization
- Early Stopping
- Làm giàu , tăng độ phức tạp của dữ liệu

Underfitting:

- Tăng độ phức tạp của mô hình
- Làm giàu , tăng độ phức tạp của dữ liệu



Thank you