

**Họ và tên:Trần Quang Đăng**

**MSSV: 52100174**

1)Thuật toán Gradient Descent cho bài toán hồi quy tuyến tính đa biến

Bước 1: Chuẩn bị dữ liệu

- Chuẩn bị dữ liệu bằng cách chia thành tập huấn luyện và tập kiểm tra.
- Chuẩn hóa dữ liệu bằng cách trừ đi trung bình và chia cho độ lệch chuẩn cho mỗi thuộc tính.

Bước 2: Khởi tạo mô hình

- Khởi tạo các tham số mô hình (trọng số và chặn) thành các giá trị ngẫu nhiên.

Bước 3: Hàm giả thuyết

- Tính đầu ra dự đoán cho mỗi ví dụ huấn luyện bằng hàm giả thuyết:  
$$h(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \dots + \theta_n * x_n$$

trong đó:

- $\theta_0$  là chặn
- $\theta_1, \theta_2, \dots, \theta_n$  là trọng số
- $x_1, x_2, \dots, x_n$  là các thuộc tính

Bước 4: Hàm mất mát

- Tính hàm mất mát, đo lường mức độ phù hợp của mô hình với dữ liệu. Hàm mất mát phổ biến cho hồi quy tuyến tính đa biến là lỗi bình phương trung bình (MSE):  
$$MSE = 1/m * \sum((h(x) - y)^2)$$

trong đó:

- $m$  là số lượng ví dụ huấn luyện
- $y$  là đầu ra thực tế

Bước 5: Gradient Descent

- Cập nhật các tham số mô hình theo hướng gradient âm của hàm mất mát. Gradient của hàm mất mát MSE cho hồi quy tuyến tính đa biến là:  
$$\text{gradient} = 1/m * \sum((h(x) - y) * x)$$

trong đó:

- $x$  là vector thuộc tính

Các tham số mô hình được cập nhật như sau:

$$\theta_i = \theta_i - \alpha * \text{gradient}_i$$

trong đó:

- $\alpha$  là tốc độ học, điều khiển bước kích thước của thuật toán gradient descent

Bước 6: Lặp lại các bước 3-5 cho đến khi hàm mất mát hội tụ hoặc đạt đến số lần lặp tối đa.

Sau khi thuật toán hội tụ, các tham số mô hình có thể được sử dụng để dự đoán đầu ra cho các điểm dữ liệu mới.

## 2) Hãy trình bày các phương pháp khác nhau để xác định tốc độ học

Có một số phương pháp khác nhau để xác định tốc độ học cho các thuật toán học máy. Một số phương pháp phổ biến nhất bao gồm:

- Thử và sai: Đây là phương pháp đơn giản nhất, nhưng cũng có thể tốn nhiều thời gian nhất. Để sử dụng phương pháp này, bạn chỉ cần bắt đầu với một tốc độ học và sau đó đào tạo mô hình của mình. Nếu mô hình không học tốt, bạn có thể thử tăng tốc độ học. Nếu mô hình học quá nhanh và overfit với dữ liệu đào tạo, bạn có thể thử giảm tốc độ học.
- Trình lập lịch tốc độ học: Trình lập lịch tốc độ học là các thuật toán tự động điều chỉnh tốc độ học trong quá trình đào tạo. Điều này có thể giúp tăng tốc quá trình đào tạo và cải thiện hiệu suất của mô hình. Một số trình lập lịch tốc độ học phổ biến bao gồm:
  - Giảm dần từng bước: Trình lập lịch này giảm tốc độ học theo một hệ số  $\gamma$  sau mỗi bước\_size epoch.
  - Giảm dần theo hàm mũ: Trình lập lịch này giảm tốc độ học theo một hệ số  $\gamma$  sau mỗi epoch.
  - Giảm dần theo hàm cosin: Trình lập lịch này giảm dần tốc độ học từ một giá trị cao xuống một giá trị thấp trong quá trình đào tạo.
- Tốc độ học thích ứng: Tốc độ học thích ứng là các thuật toán điều chỉnh tốc độ học cho từng tham số của mô hình. Điều này có thể giúp cải thiện hiệu suất của mô hình trên các tác vụ phức tạp. Một số thuật toán tốc độ học thích ứng phổ biến bao gồm:
  - Adam: Thuật toán này là lựa chọn phổ biến cho nhiều tác vụ học máy. Nó dựa trên thuật toán ước tính mômen thích ứng (Adam).
  - RMSprop: Thuật toán này là lựa chọn phổ biến khác cho các tác vụ học máy. Nó dựa trên thuật toán truyền bá phương sai bình phương gốc (RMSprop).