

Tạo một colab với tên là “ho\_va\_ten\_ML1.ipynb” để thực hiện các việc sau. Nộp [link chia sẻ colab](#) và bản [in .pdf](#) của colab.

### Bài 1

1. Import thư viện numpy với tên rút gọn là ‘np’
2. Import thư viện matplotlib.pyplot với tên rút gọn là ‘plt’
3. Dùng hàm ‘np.linspace’ để tạo ra numpy array ‘x’ trong khoảng [-3, 3] với 100 giá trị
4. Tạo ra chuỗi numpy array ‘y’ chứa giá trị của hàm:  $f(x) = \sin(2x)(\frac{x^3}{5} + x^2 + 5x)$
5. Dùng hàm ‘plt.plot(x, y)’ vẽ đồ thị
6. Định nghĩa một hàm ‘f’ nhận vào giá trị một giá trị x trả lại giá trị hàm f(x) trong mục 4.
7. Định nghĩa hàm ‘grad\_f’ trả lại giá trị đạo hàm của hàm f(x) tại một giá trị của x
8. Chọn xo = -1.5, training rate = 0.01, dùng thuật toán gradient descent để tìm cực trị của hàm f(x)
9. Lặp lại mục 8 với xo = 0.5, và xo = 2.0. Cho nhận xét

**Bài 2.** Thực hiện một nghiên cứu nhằm xác định thu nhập của người lao động theo số năm kinh nghiệm làm việc, người ta thu được bản số liệu như sau:

Kinh nghiệm làm việc (năm)	1	2	3	4	5
Thu nhập (nghìn USD/năm)	39,3	43,5	56,6	67,9	73,1

Giả sử mức thu nhập phụ thuộc tuyến tính vào kinh nghiệm như sau:  $\hat{y} = a \times x + b$ , trong đó  $\hat{y}$  là giá trị dự đoán mức thu nhập của người lao động (nghìn USD/năm), x là kinh nghiệm làm việc (năm), a và b là các thông số xác định mô hình dự đoán.

1. Tính giá trị trung bình, độ lệch chuẩn của thu nhập
2. Tính hệ số tương quan của số năm kinh nghiệm và mức thu nhập và cho nhận xét.
3. Viết biểu thức tính loss function (L) phụ thuộc vào a và b:

$$L(a, b) = \sum_i \frac{1}{2} (\hat{y}_i - y_i)^2, \text{ với } \hat{y}_i = a \times x_i + b.$$

4. Viết hàm tính đạo hàm của L theo a và b. Tìm giá trị a và b, để cho hàm L nhỏ nhất bằng thuật toán gradient descent.
5. Hỏi một sinh viên mới tốt nghiệp thì nên đề xuất mức lương là bao nhiêu
6. Để mức lương tăng lên 10000 thì một SV mới tốt nghiệp phải tích lũy thêm bao nhiêu năm kinh nghiệm