**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**

**-----------------⸙∆⸙-----------------**



**ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI:**

**“Bài tập 1”**

**GVHD: Trần Vũ Hoàng**

**SVTH: Nguyễn Công Trung**

**MSSV: 2341109**

**Tp. Hồ Chí Minh tháng 06 năm 2024**

# Mục lục

[Mục lục i](#_Toc170551564)

[Danh sách hình ảnh ii](#_Toc170551565)

[Chương 1. Đề bài 1](#_Toc170551566)

[Chương 2. BÀI LÀM 2](#_Toc170551567)

[2.1 Code 2](#_Toc170551568)

[2.2 Kết quả 3](#_Toc170551569)

[Chương 3. KẾT LUẬN 7](#_Toc170551570)

# Danh sách hình ảnh

[Hình 1. Learning rate = 1, số vòng lặp = 100 3](#_Toc170551556)

[Hình 2. Learning rate = 0.1, số vòng lặp = 100 3](#_Toc170551557)

[Hình 3. Learning rate = 0.01, số vòng lặp = 100 4](#_Toc170551558)

[Hình 4. Learning rate = 0.001, số vòng lặp = 100 4](#_Toc170551559)

[Hình 5. Learning rate = 0.0001, số vòng lặp = 100 5](#_Toc170551560)

[Hình 6. Learning rate = 0.0001, số vòng lặp = 10 5](#_Toc170551561)

[Hình 7. Learning rate = 0.00001, số vòng lặp = 10 6](#_Toc170551562)

[Hình 8. Learning rate = 0.00001, số vòng lặp = 10 6](#_Toc170551563)

# Đề bài

Trong file data1.csv (utex) có lưu dữ liệu x và y cho bài toán hồi quy tuyến tính với 1 biến. Các bạn làm một số yêu cầu sau:

• Nhập dữ liệu vào và visualize dữ liệu ra, lưu trữ x và y riêng biệt thành các ma trận tương ứng.

• Viết hàm cập nhật giá trị theta 0 và theta 1 sử dụng phương pháp gradient descent với số lượng vòng lặp và learning rate (alpha) tùy chọn.

• Viết hàm tính cost function J ở mỗi vòng lặp.

• Đánh giá tác động của số lượng vòng lặp và các giá trị của learning rate vào hàm cost. Có nghĩa là các bạn phải vẽ sự thay đổi của J sau mỗi vòng lặp ở các giá trị learning rate nhất định.

• Biểu diễn đường thẳng cuối cùng tìm thấy và các điểm dữ liệu trên cùng một trục tọa độ.

• Các bạn phải làm các bước trên và copy ảnh vào trong 1 file word để nộp lên.

# BÀI LÀM

## Code

import polars as pl

import numpy as np

import matplotlib.pyplot as plt

alpha = 0.0001

df = pl.read\_csv('data.csv')

x = df.select('x').to\_numpy().reshape(-1, 2)

x\_mat = df.select(pl.lit(1).alias('ones'), 'x').to\_numpy().reshape(-1, 2)

y = df.select('y').to\_numpy().reshape(-1, 1)

m = len(y)

theta = np.random.rand(2, 1)

H = np.matmul(x\_mat, theta)

loop = 10

J\_values = np.zeros((loop, 1))

for i in range(loop):

theta = theta - (alpha/m)\*np.matmul(x\_mat.T, (H-y))

H = np.matmul(x\_mat, theta)

J\_values[i] = (1/(2\*m))\*np.matmul((H-y).T, (H-y))

# Create a figure and a set of subplots

fig, axs = plt.subplots(2)

# Plot the cost function on the first subplot

axs[0].plot(J\_values)

axs[0].set\_xlabel('Iteration')

axs[0].set\_ylabel('Cost (J)')

axs[0].set\_title('Cost Function')

# Plot the line and (x, y) on the second subplot

axs[1].scatter(x, y)

x\_line = np.linspace(x.min(), x.max(), 100)

y\_line = theta[1]\*x\_line + theta[0]

axs[1].plot(y\_line, 'r-')

axs[1].set\_xlabel('x')

axs[1].set\_ylabel('y')

axs[1].set\_title('Function')

# Layout so plots do not overlap

fig.tight\_layout()

plt.show()

## Kết quả

Thông số learning rate = 1, số vòng lặp = 100.

A graph of a function

Description automatically generated

Hình . Learning rate = 1, số vòng lặp = 100

Thông số learning rate = 0.1, số vòng lặp = 100.

A graph of function and function

Description automatically generated

Hình . Learning rate = 0.1, số vòng lặp = 100

Thông số learning rate = 0.01, số vòng lặp = 100.

A graph of function and cost function

Description automatically generated

Hình . Learning rate = 0.01, số vòng lặp = 100

Thông số learning rate = 0.001, số vòng lặp = 100.

A graph of function and function

Description automatically generated

Hình . Learning rate = 0.001, số vòng lặp = 100

Thông số learning rate = 0.0001, số vòng lặp = 100.

A graph of function and function

Description automatically generated with medium confidence

Hình . Learning rate = 0.0001, số vòng lặp = 100

Thông số learning rate = 0.0001, số vòng lặp = 10.

A graph of a function

Description automatically generated

Hình . Learning rate = 0.0001, số vòng lặp = 10

Thông số learning rate = 0.00001, số vòng lặp = 10.

A graph of a function

Description automatically generated

Hình . Learning rate = 0.00001, số vòng lặp = 10

Thông số learning rate = 0.00001, số vòng lặp = 10.

A graph of function and function

Description automatically generated

Hình . Learning rate = 0.00001, số vòng lặp = 10

# KẾT LUẬN

Thông số alpha quan trọng trong việc để kết quả có thể hội tự và tránh bị phân kì. Với số lượng vòng lặp khác nhau , kết quả sẽ cho ra chính xác hơn và hội tụ nhanh hơn.