BÀI TẬP TUẦN 1

Võ Xuân Diệu – CAMERA AI

1. Linear Regression:

Ví dụ ứng dụng thực tế:

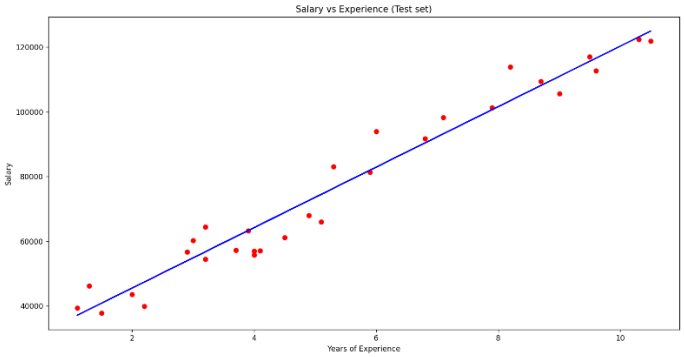
- Rút ra mối quan hệ tuyến tính giữa lương và số năm kinh nghiệm (Simple Linear Regression).

- Dữ liệu đầu vào:

  + 31 điểm dữ liệu

  + Feature: YearsExperience

  + Label: Salary



Code:

```
# Linear Regression Đơn giản
# Nhập thư viện
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Nhập dữ liệu đầu vào
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

# Chưa dữ liệu ra làm bộ huấn luyện và bộ thử
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)

# Sử dụng thư viện scikit-learn để tìm best fitting line
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Thử trên tập dữ liệu thử
y_pred = regressor.predict(X_test)

# Vẽ ra fitting line
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')

# Vẽ ra dữ liệu thử
```

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

2. K-mean Clustering

Ví dụ ứng dụng thực tế:

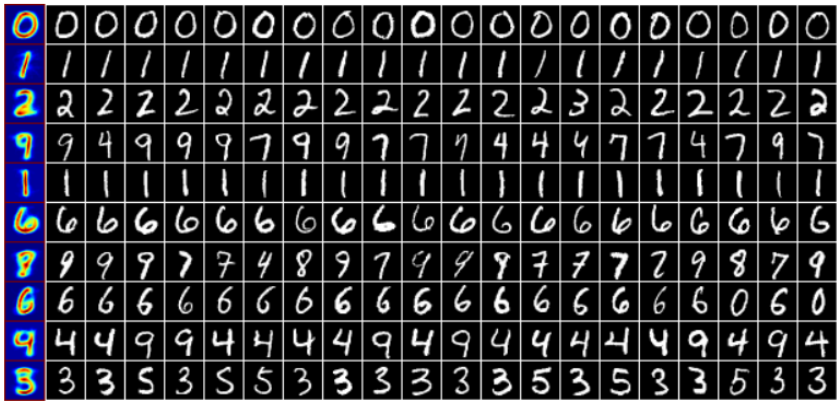
- Phân loại chữ số viết tay trên các bưu kiện

- Dữ liệu đầu vào:

  + Bộ dữ liệu MNIST, bộ dữ liệu lớn chứa hình ảnh của chữ số viết tay.

  + 60,000 ví dụ trong tập huấn luyện, 10,000 ví dụ trong tập thử.

  + Mỗi ví dụ là một hình ảnh grayscale của chữ số viết tay (từ 1 đến 9), có kích thước 28x28 pixel.



Áp dụng K-means clustering vào tập test set của bộ cơ sở dữ liệu MNIST với K = 10 cluster. Cột 1: centers của các cluster. Các cột còn lại: Mỗi hàng là 20 điểm dữ liệu gần center nhất của mỗi cluster.

3. K-nearest Neighbor:

Ví dụ ứng dụng thực tế:

- Phân loại Hoa Iris vào một trong ba loại khác nhau: Iris setosa, Iris Versicolor và Iris virginica.

- Dữ liệu đầu vào:

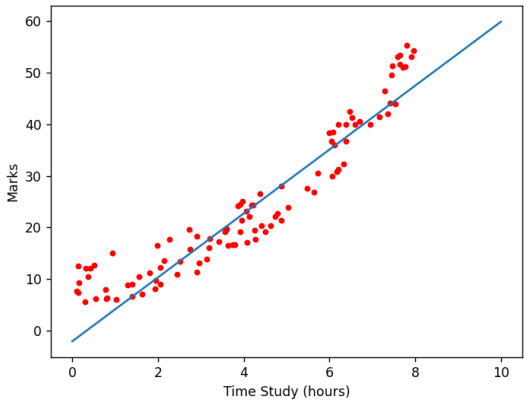
  + Bộ dữ liệu hoa Iris.

+ Gồm 150 điểm dữ liệu chứa thông tin của ba dạng hoa Iris (chiều dài, chiều rộng đài hoa (sepal), và chiều dài, chiều rộng cánh hoa (petal)).

4. Gradient Descent:

Ví dụ ứng dụng thực tế:

- Áp dụng thuật toán Gradient Descent trong bài toán Linear Regression để tìm mối liên hệ tuyến tính giữa điểm cả học sinh và khoảng thời gian học.
- Dữ liệu đầu vào:
  + Label: Giá trị điểm của học sinh.
  + Feature: Thời gian học, số môn học
  + Bộ dữ liệu có 101 mẫu.



Code:

```
from __future__ import division, print_function, unicode_literals
import pandas as pd
import numpy as np
import math
from scipy import sparse
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# Nhập dữ liệu Boston
file_name = 'Student_Marks.csv'
dataFrame = pd.read_csv(file_name)

# Phân loại giữa biến Label và biến Feature
Label=['Marks']
Feature=['number_courses','time_study']
y = dataFrame[Label].values
X = dataFrame[Feature].values

# Chia tập dữ liệu thành 2 tập : Tập huấn luyện và tập thử nghiệm
one = np.ones((X.shape[0],1))
X = np.concatenate((one, X), axis = 1)

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)

N1 = X_train.shape[0]
N2 = X_test.shape[0]

# Hàm mất mát
def cost(W1,N,Y,X):
    return .5/N*(np.linalg.norm(Y - np.dot(X,W1), 2))**2;
```

```
# Gradient
def gradient(W1,N):
    return 1/N*(X_train.T).dot(X_train.dot(W1) - y_train)

d0 = X.shape[1]
d1 = C = 1
# Khởi tạo giá trị random cho các trọng số
W1 = np.random.randn(d0, d1)

eta = 0.01# Tốc độ học
for                                                    i                                                    in
range(100):

    ## tính MSE
    loss = cost(W1,N1,y_train,X_train)
    # In ra Mean Square Error mỗi 1000 vòng lặp
    if i %10 == 0:
        print("iter %d, loss: %f" %(i, loss))

    # backpropagation
    dW1=gradient(W1,N1)

    # Cập nhật trọng số
    W1 = W1 -eta*dW1

loss = cost(W1,N2,y_test,X_test)
print("Testing:  MSE: %f" %( loss))

x1 = x2 = np.linspace(0, 10, 2, endpoint=True)

line = W1[0] + W1[1]*x1 + W1[2]*x2;


# Hiển thị kết quả dưới dạng biểu đồ
plt.plot(X[:,2],y, 'r.', markersize = 7);
plt.plot(x2,line);
plt.xlabel("Time Study (hours)")
plt.ylabel("Marks")
plt.show()
```

- Nguồn tham khảo: https://machinelearningcoban.com/