

Object Oriented Programming - Day 11

Ngày 21 tháng 6 năm 2024

Ngày thực hiện:	21/06/2024
Người thực hiện:	Đinh Thị Tâm
Nguồn:	AIO2024 - Day 11
Nguồn dữ liệu (nếu có):	Link Sources code
Từ khóa:	Object Oriented Programming
Người tóm tắt:	Đinh Thị Tâm

1. Mô tả

Lập trình hướng đối tượng (OOP) là một phương pháp lập trình tập trung vào việc tạo ra các "đối tượng", là những đơn vị cơ bản đại diện cho các thực thể trong thế giới thực. Mỗi đối tượng có các thuộc tính (data) và phương thức (hành vi) riêng. OOP mang lại nhiều lợi ích cho việc lập trình, bao gồm:

- Tái sử dụng mã: OOP cho phép bạn viết mã có thể được sử dụng lại nhiều lần cho các mục đích khác nhau. Điều này giúp tiết kiệm thời gian và công sức, đồng thời giúp mã dễ dàng bảo trì hơn.
- Tính linh hoạt: OOP giúp bạn dễ dàng thay đổi và mở rộng mã của mình. Khi bạn cần thêm tính năng mới, bạn chỉ cần tạo ra các đối tượng mới hoặc sửa đổi các đối tượng hiện có.
- Tính bảo trì: OOP giúp mã của bạn dễ đọc và dễ hiểu hơn. Điều này giúp bạn dễ dàng tìm và sửa lỗi, đồng thời giúp những người khác dễ dàng hiểu mã của bạn.

2. Bài tập:

Tạo lớp `ReLUActivateFunction`, `SigmoidActivateFunction` được kế thừa từ lớp `nn.Module` của thư viện PyTorch. Tiến hành cài đặt hai lớp trên và thực nghiệm với `tensor = [1, -5, 1.5, 2.7, -5]`. Biết công thức của ReLU và sigmoid như sau:

$$\text{ReLU} = \max(0, x)$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-x}}$$

(a) Code

```

1 import torch.nn as nn
2 import math
3
4
5 class ReLUActivateFunction(nn.Module):
6     def __init__(self, x):
7         self.__x = x

```

```
8
9     def __call__(self):
10         a = []
11         length = len(self.__x)
12         for idx in range(length):
13             if (self.__x[idx] > 0):
14                 a.append(self.__x[idx])
15             else:
16                 a.append(0)
17         return a
18
19
20 class SigmoidActivateFunction(nn.Module):
21     def __init__(self, x):
22         self.__x = x
23
24     def __call__(self):
25         a = []
26         length = len(self.__x)
27         for idx in range(length):
28             a.append(round(1/(1+math.exp(-self.__x[idx])), 4))
29         return a
30
31
32     # test
33 Relua = ReLUActivateFunction([1, -5, 1.5, 2.7, -5])
34 print(f'Relua: {Relua()}')
35 Sigmoid = SigmoidActivateFunction([1, -5, 1.5, 2.7, -5])
36 print(f'Sigmoid: {Sigmoid()}')
```

(b) Kết quả

Relua: [1, 0, 1.5, 2.7, 0]

Sigmoid: [0.7311, 0.0067, 0.8176, 0.937, 0.0067]