

实验 03 循环与选择结构

实验内容: 通过自拟案例, 初始化相关变量, 设置简易的代码框架, 测试使用 while 循环, for 循环, if 等选择结构和相关的关键字与函数. 完成思考题.

实验过程:

- 1) 初始化变量并测试: 测试使用 while 和 for 循环, 实现 for/while 循环的(两层)嵌套使用.
- 2) 熟悉与练习关键字 break 和 continue. 实现设置"while True:"循环, 设置退出循环的测试条件.
- 3) 练习使用 enumerate(), zip(), range(), input()等函数. 实现分别用 enumerate() 和 range(), 在循环中打印出"目前迭代第 i 个元素"字符串, 字符串中" i "为当前迭代元素的索引值或迭代序号(字典没有索引).
- 4) 测试使用 if 结构, if-else 结构, if-elif-else 结构. 实现 Leaky ReLU 函数并完成以下任务(中文:"带泄露线性整流函数"; 神经网络中常用激活函数, 自行百度/Google 一下定义和相关背景):
 - 4.1 实现一个具有 if-else 结构的 Leaky ReLU 函数, 命名和参数为 "leaky_relu_v1(x, a=0.2)", 这里 a 为 x 为负值时的系数.
 - 4.2 实现一个没有 if-else 结构的 Leaky ReLU 函数, 命名和参数为 "leaky_relu_v2(x, a=0.2)".
 - 4.3 通过 for 循环计算-10000000 到 10000000 的 Leaky ReLU 函数值, 并记录总计算时间(这里可用 time 模块中的 time()函数).
 - 4.4 打印和比较函数的两个不同实现的总计算时间开销.
- 5)* 了解对序列反向迭代的用法.
- 6)* 了解 itertools 模块, 与其所包含的各类迭代器工具, 包括 chain(), cycle(), islice(), product(), permutations(), combinations()等.
- 7)* 了解 iter()函数的运作机制.

带"*"项可不用写入实验报告

回答思考题:

1. 为什么循环结构中不应改变正在迭代的序列的长度? 能否用代码写出一个反例?
2. 为什么程序设计语言中需要有循环结构和选择结构?

相关参考:

[1] 百度百科, ReLU 函数:

<https://baike.baidu.com/item/ReLU%20%E5%87%BD%E6%95%B0/22689567>