

# 簡單的Python調用C++程序

機器學習算法那些事 2021-12-26 12:58

## Python調用C/C++程序的方法

最近寫BUG的時候遇到python計算很慢的情況，於是調研了一波在python中嵌入C++程序的方法，記錄一下，便於查詢。

一般來說在python調用C/C++程序主要可以分為3步：

- 1、編寫C/C++實現程序。 - 2、將C/C++程序編譯成動態庫。 - 3、在Python中調用編譯生成的庫。Python在調用C/C++程序時有一些不同，需要注意。

### 1、Python調用C函數

Python調用C語言程序比較簡單，將C語言程序編譯好，再使用python中的ctypes模塊調用即可。

**C語言源文件：called\_c.c**

```
//编译命令 gcc -o libpycall.so -shared -fPIC called_c.c
#include<stdio.h>
int foo(int a, int b){<!-- -->
printf("a:%d, b:%d.", &a, &b);
return 0;
}
```

在命令行或者終端輸入：

```
gcc -o libpycall.so -shared -fPIC called_c.c
```

生成libpycall.so動態庫文件，之後就可以在Python中調用foo函數。**Python文件：py\_call\_c.py**

```
import ctypes
dll = ctypes.cdll.LoadLibrary
lib = dll('./libpycall.so') //刚刚生成的库文件的路径
lib.foo(1, 3)
```

運行py\_call\_c.py輸出為：

```
a:1, b:3
```

## 2、Python調用C++類

由於C++支持函數重載，在g++以C++方式編譯時編譯器會給函數的名稱附加上額外的信息，這樣ctypes模塊就會找不到g++編譯生成的函數。因此，要讓g++按照C語言的方式編譯才可以找到生成的函數名。讓編譯器以C語言的方式編譯就要在代碼中使用extern關鍵字將代碼包裹起來。

### C++源文件:cpp\_called.cpp

```
//Python调用c++(类) 动态链接库
#include <iostream>
using namespace std;

class TestLib
{<!-- -->
public:
    void display();
    void display(int a);
};

void TestLib::display() {<!-- -->
    cout<<<"First display"<<<endl;
}

void TestLib::display(int a) {<!-- -->
    cout<<<"Second display:"<<<a<<<endl;
}
```

```
extern "C" {<!-- -->
    TestLib obj;
    void display() {<!-- -->
        obj.display();
    }
    void display_int(int a) {<!-- -->
        obj.display(a);
    }
}
```

在命令行或者終端輸入編譯命令：

```
g++ -o libpycallcpp.so -shared -fPIC cpp_called.cpp
```

編譯參數說明-fPIC：生成位置無關目標代碼，適用於動態連接；-L path：表示在path目錄中搜索庫文件，如-L.表示在當前目錄；-I path：表示在path目錄中搜索頭文件；-o file：制定輸出文件為file；-shared：生成一個共享庫文件；

生成libpycallcpp.so，在Python中調用。**Python文件：py\_call\_c.py**

```
import ctypes
dll = ctypes.cdll.LoadLibrary
lib = dll('./libpycallcpp.so') //刚刚生成的库文件的路径
lib.display()
lib.display_int(0)
```

輸出為：

```
First display
Second display:0
```

OK，基本功能實現，高級調用後期在填坑

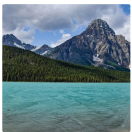
— The End —



喜歡此內容的人還喜歡

Python的十大特性

AI前線



Python 200個標準庫匯總

小白學視覺

