

Assignment #2: 深度学习与大语言模型

Updated 2204 GMT+8 Feb 25, 2025

2025 spring, Compiled by 胡新璞, 工学院

1. 题目

18161: 矩阵运算

matrices, <http://cs101.openjudge.cn/practice/18161>

思路：按矩阵老老实实做就行

代码：

```
a_row,a_col = map(int,input().split())
matrix_a = [0] * a_row
for _ in range(a_row):
    matrix_a[_] = list(map(int, input().split()))
b_row,b_col = map(int,input().split())
matrix_b = [0] * b_row
for _ in range(b_row):
    matrix_b[_] = list(map(int, input().split()))
c_row,c_col = map(int,input().split())
matrix_c = [0] * c_row
for _ in range(c_row):
    matrix_c[_] = list(map(int, input().split()))
if a_col != b_row or a_row != c_row or b_col != c_col:
    print("Error!")
else:
    matrix_ab,matrix = [0] * a_row,[0] * a_row
    for _ in range(a_row):
        matrix_ab[_],matrix[_] = [0] * b_col,[0] * b_col
    for i in range(a_row):
        for j in range(b_col):
            for k in range(a_col):
                matrix_ab[i][j] += matrix_a[i][k] * matrix_b[k][j]
            matrix[i][j] = matrix_ab[i][j] + matrix_c[i][j]
    for i in range(a_row):
        print(" ".join(map(str,matrix[i])))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
a_row,a_col = map(int,input().split())
matrix_a = [0] * a_row
```

基本信息

#: 48511283

题目: 18161

提交人: 2400011037

19942: 二维矩阵上的卷积运算

matrices, <http://cs101.openjudge.cn/practice/19942/>

思路：只要读懂题目就会做了（）

代码：

```
m,n,p,q = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(m)]
matrix_ = [list(map(int, input().split())) for _ in range(p)]
res = [[0] * (n + 1 - q) for _ in range(m + 1 - p)]
for i in range(m + 1 - p):
    for j in range(n + 1 - q):
        for k in range(p):
            for l in range(q):
                res[i][j] += matrix[i + k][j + l] * matrix_[k][l]
for i in range(m + 1 - p):
    print(" ".join(map(str, res[i])))
```

代码运行截图（至少包含有"Accepted"）

状态: **Accepted**

源代码

```
m,n,p,q = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(m)]
matrix_ = [list(map(int, input().split())) for _ in range(p)]
res = [[0] * (n + 1 - q) for _ in range(m + 1 - p)]
for i in range(m + 1 - p):
    for j in range(n + 1 - q):
        for k in range(p):
            for l in range(q):
                res[i][j] += matrix[i + k][j + l] * matrix_[k][l]
for i in range(m + 1 - p):
    print(" ".join(map(str, res[i])))
```

基本信息

#: 48511313
题目: 19942
提交人: 2400011037
内存: 3660kB
时间: 30ms
语言: Python3
提交时间: 2025-03-10 15:44:12

04140: 方程求解

牛顿迭代法, <http://cs101.openjudge.cn/practice/04140/>

请用牛顿迭代法实现。

因为大语言模型的训练过程中涉及到了梯度下降（或其变种，如 SGD、Adam 等），用于优化模型参数以最小化损失函数。两种方法都是通过迭代的方式逐步接近最优解。每一次迭代都基于当前点的局部信息调整参数，试图找到一个比当前点更优的新点。理解牛顿迭代法有助于深入理解基于梯度的优化算法的工作原理，特别是它们如何利用导数信息进行决策。

代码：

```
def Newton(x):  
    xx = x - (x ** 3 - 5 * x ** 2 + 10 * x - 80) / (3 * x ** 2 - 10 * x + 10)  
    return xx  
  
x = float(6)  
for i in range(10):  
    x = Newton(x)  
  
print('%.9f' % x)
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
def Newton(x):  
    xx = x - (x ** 3 - 5 * x ** 2 + 10 * x - 80) / (3 * x ** 2 - 10 * x +  
    return xx  
  
x = float(6)  
for i in range(10):  
    x = Newton(x)  
  
print('%.9f' % x)
```

基本信息

#: 48511491
题目: 04140
提交人: 2400011037
内存: 3588kB
时间: 34ms
语言: Python3
提交时间: 2025-03-10 15:57:23

06640: 倒排索引

data structures, <http://cs101.openjudge.cn/practice/06640/>

思路：用的比较简单直接的写法来找，字典存储文档

代码：

```
n = int(input())
data = {}
for _ in range(1,n+1):
    lst = list(map(str, input().split()))
    lst.pop(0)
    data[_] = lst
m = int(input())
for _ in range(m):
    word = input()
    word_index = []
    for i in range(1,n+1):
        if word in data[i]:
            word_index.append(i)
    print(" ".join(map(str,word_index))) if word_index else print("NOT FOUND")
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
n = int(input())
data = {}
for _ in range(1,n+1):
    lst = list(map(str, input().split()))
    lst.pop(0)
    data[_] = lst
m = int(input())
for _ in range(m):
    word = input()
    word_index = []
    for i in range(1,n+1):
        if word in data[i]:
            word_index.append(i)
    print(" ".join(map(str,word_index))) if word_index else print("NOT FOUND")
```

基本信息

#: 48511887
题目: 06640
提交人: 2400011037
内存: 9264kB
时间: 953ms
语言: Python3
提交时间: 2025-03-10 16:17:31

04093: 倒排索引查询

data structures, <http://cs101.openjudge.cn/practice/04093/>

思路：自己的思路跟倒排索引类似，但是非常暴力，只能 pypy3 过。。。看了题解学习了集合的运算。

代码：

```
n = int(input())    #n 个词，以及它们所处的文档编号
dict_word = {}
document = []
for _ in range(1,n+1):
    lst = list(map(int,input().split()))
    lst.pop(0)
    dict_word[_] = lst
    document.extend(lst)

document = list(set(document))
document.sort()

m = int(input())
for _ in range(m):    #m 组查询条件
    lst1 = list(map(int,input().split()))
    ans = []
    for i in range(len(document)):    #针对每一个文档进行确认
        flag = True
        for j in range(1,n+1):    #针对每一个词，判断是否符合题意
            if lst1[j-1] == 1 and (not document[i] in dict_word[j]):
                flag = False
            elif lst1[j-1] == -1 and (document[i] in dict_word[j]):
                flag = False
        if flag:
            ans.append(document[i])
    ans.sort()
    print(" ".join(map(str,ans))) if ans else print("NOT FOUND")
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
n = int(input())    #n 个词，以及它们所处的文档编号
dict_word = {}
document = []
for _ in range(1,n+1):
    lst = list(map(int,input().split()))
    lst.pop(0)
    dict_word[_] = lst
    document.extend(lst)
```

基本信息

#: 48517328
题目: 04093
提交人: 2400011037
内存: 81612kB
时间: 709ms
语言: PyPy3
提交时间: 2025-03-10 23:55:28

Q6. Neural Network 实现鸢尾花卉数据分类

在 <http://clab.pku.edu.cn> 云端虚拟机，用 Neural Network 实现鸢尾花卉数据分类。

参考链接，https://github.com/GMyhf/2025spring-cs201/blob/main/LLM/iris_neural_network.md

2. 学习总结和个人收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算 2025spring 每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

还在找手感中。对倒排索引这样的题目，自己的那种思路已经显得有点麻烦了，在学习题解的思路中。