Assignment #4: 位操作、栈、链表、堆和NN

Updated 1203 GMT+8 Mar 10, 2025

2025 spring, Complied by 任宇桐 物理学院

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
>
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

136.只出现一次的数字

bit manipulation, https://leetcode.cn/problems/single-number/

请用位操作来实现，并且只使用常量额外空间。

看了题解，觉得位操作的方式实在很神奇，学到了。

代码：

```
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        ans = nums[0]
        for i in nums[1:]:
            ans ^= i
        return ans
```

代码运行截图 （至少包含有"Accepted"）

20140:今日化学论文

stack, http://cs101.openjudge.cn/practice/20140/

思路:

看起来思路比较简单，但是还是debug了1h左右……注意到要判断输入是数字的一部分还是字符串，同时不管是遇到字母还是前括号都要将数字压入栈（会有[x[xs]]的情形……）

代码:

```
s = input()
num = ''
temp = ''
stack = []
for i in range(len(s)):
    if s[i].isalpha() or s[i] == '[':
        if num != '':
            stack.append(int(num))
            num = ''
        stack.append(s[i])
    elif s[i].isnumeric():
        num += s[i]
    elif s[i] == ']':
        temp = ''
        while not isinstance(stack[-1], int) and stack[-1] != '[':
            temp = stack.pop() + temp
        if stack[-1] != '[':
            temp = stack.pop() * temp
        stack.pop()
        stack.append(temp)
print(*stack, sep='')
```

代码运行截图 （至少包含有"Accepted"）

160.相交链表

linked list, https://leetcode.cn/problems/intersection-of-two-linked-lists/

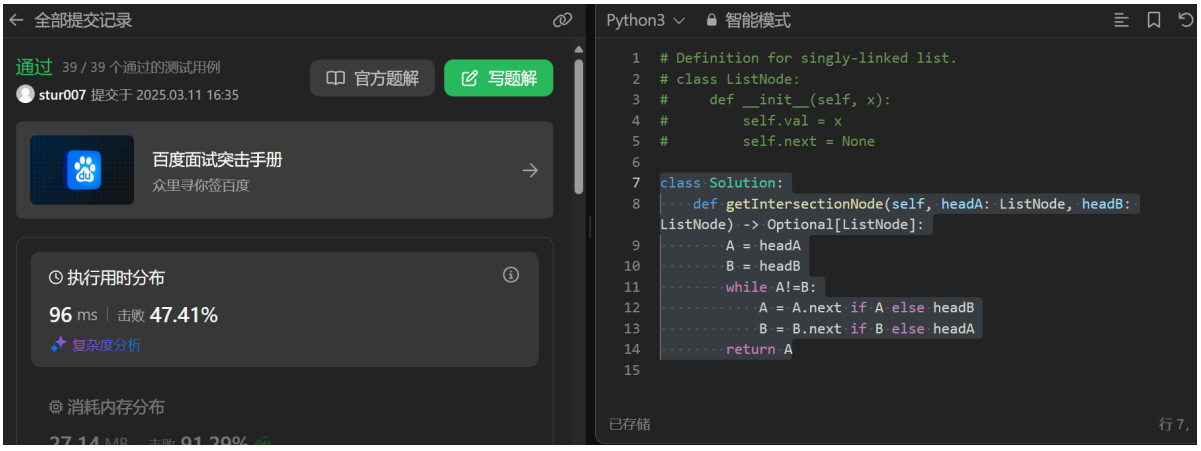
思路:

开始自己写只会hash表，参考了题解以后学到了双指针的方法，实在是简单。

代码:

```
class Solution:
    def getIntersectionNode(self, headA: ListNode, headB: ListNode) ->
Optional[ListNode]:
        A = headA
        B = headB
        while A!=B:
            A = A.next if A else headB
            B = B.next if B else headA
        return A
```

代码运行截图 （至少包含有"Accepted"）



206.反转链表

linked list, https://leetcode.cn/problems/reverse-linked-list/

思路：

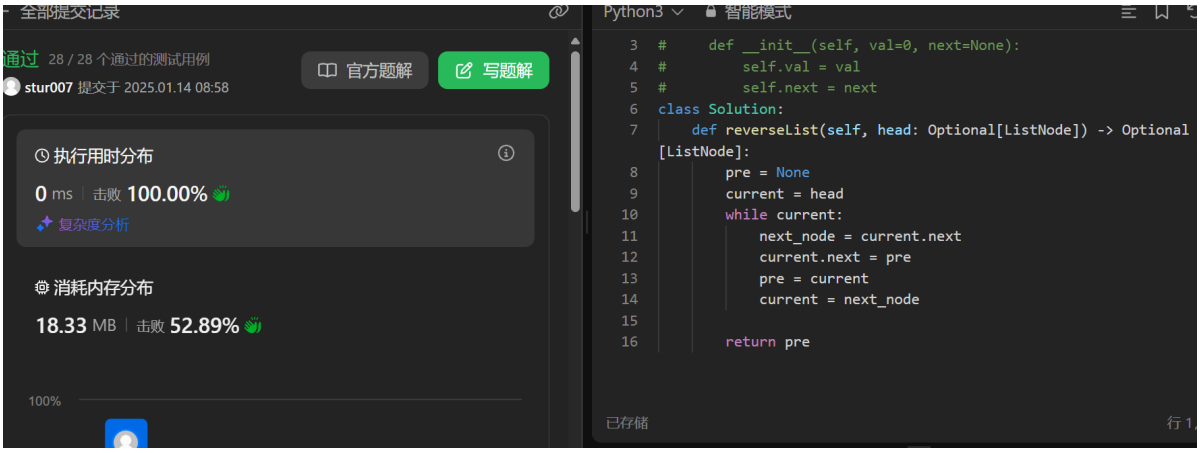
在链表处修改指针的指向即可。

代码：

```python
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        pre = None
        current = head
        while current:
            next_node = current.next
            current.next = pre
            pre = current
            current = next_node

        return pre
```

代码运行截图 （至少包含有"Accepted"）



3478.选出和最大的K个元素

heap, https://leetcode.cn/problems/choose-k-elements-with-maximum-sum/

思路：

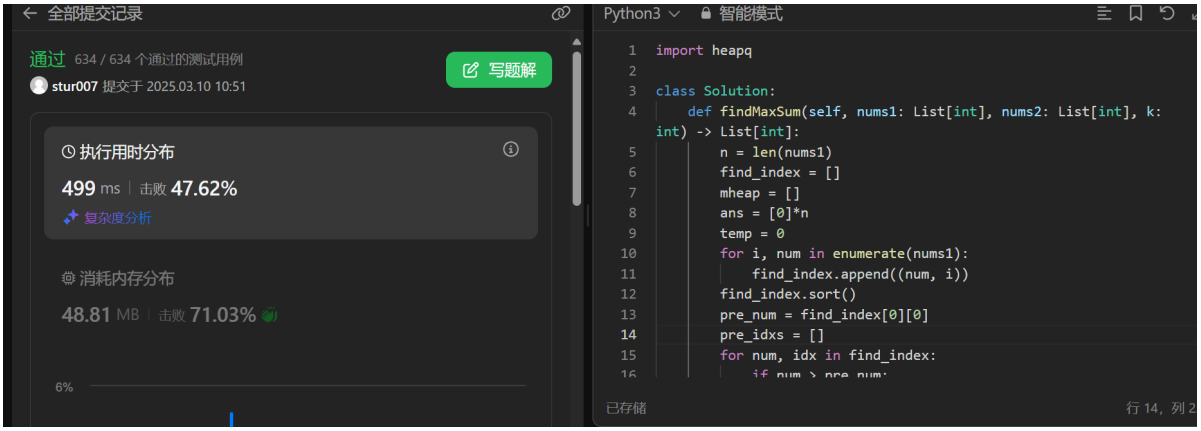
一个列表用来排序，一个列表用来求值。

代码：

```python
import heapq
```

```python
class Solution:
    def findMaxSum(self, nums1: List[int], nums2: List[int], k: int) ->
List[int]:
        n = len(nums1)
        find_index = []
        mheap = []
        ans = [0]*n
        temp = 0
        for i, num in enumerate(nums1):
            find_index.append((num, i))
        find_index.sort()
        pre_num = find_index[0][0]
        pre_idxs = []
        for num, idx in find_index:
            if num > pre_num:
                pre_num = num
                while pre_idxs:
                    pre_idx = pre_idxs.pop()
                    temp += nums2[pre_idx]
                    heapq.heappush(mheap, nums2[pre_idx])
                while len(mheap)>k:
                    temp -= heapq.heappop(mheap)
            pre_idxs.append(idx)
            ans[idx] = temp
        return ans
```

代码运行截图 （至少包含有"Accepted"）



Q6.交互可视化neural network

https://developers.google.com/machine-learning/crash-course/neural-networks/interactive-exercises

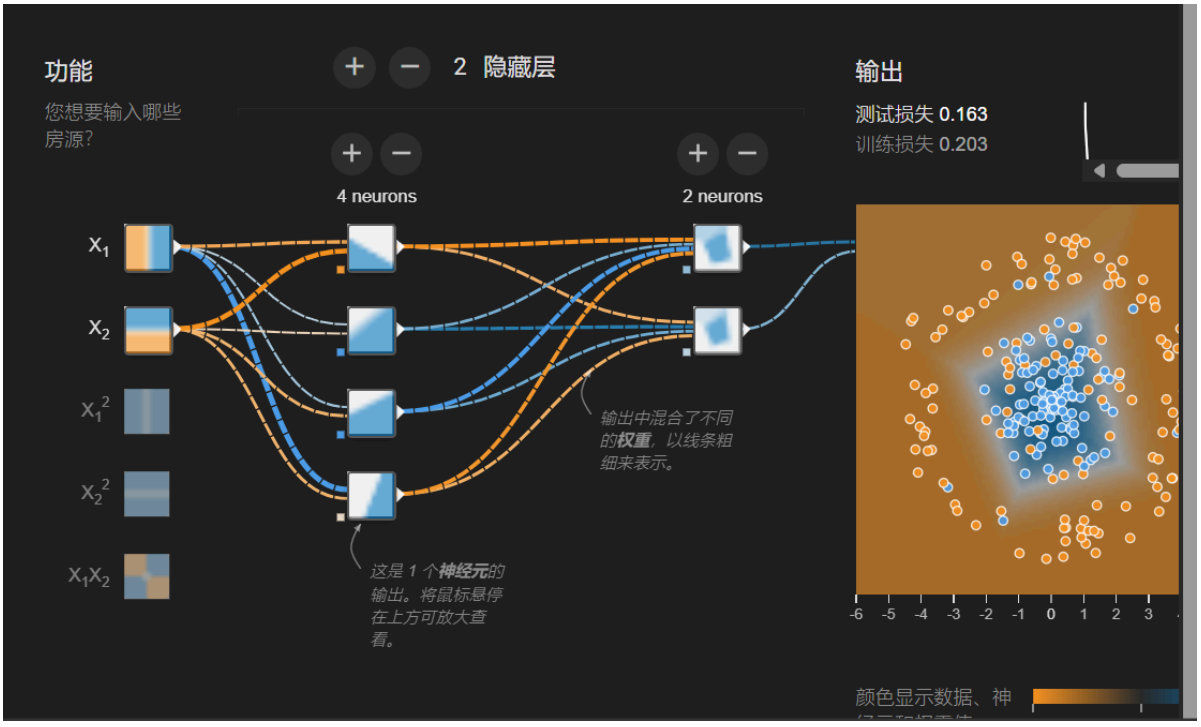
Your task: configure a neural network that can separate the orange dots from the blue dots in the diagram, achieving a loss of less than 0.2 on both the training and test data.

Instructions:

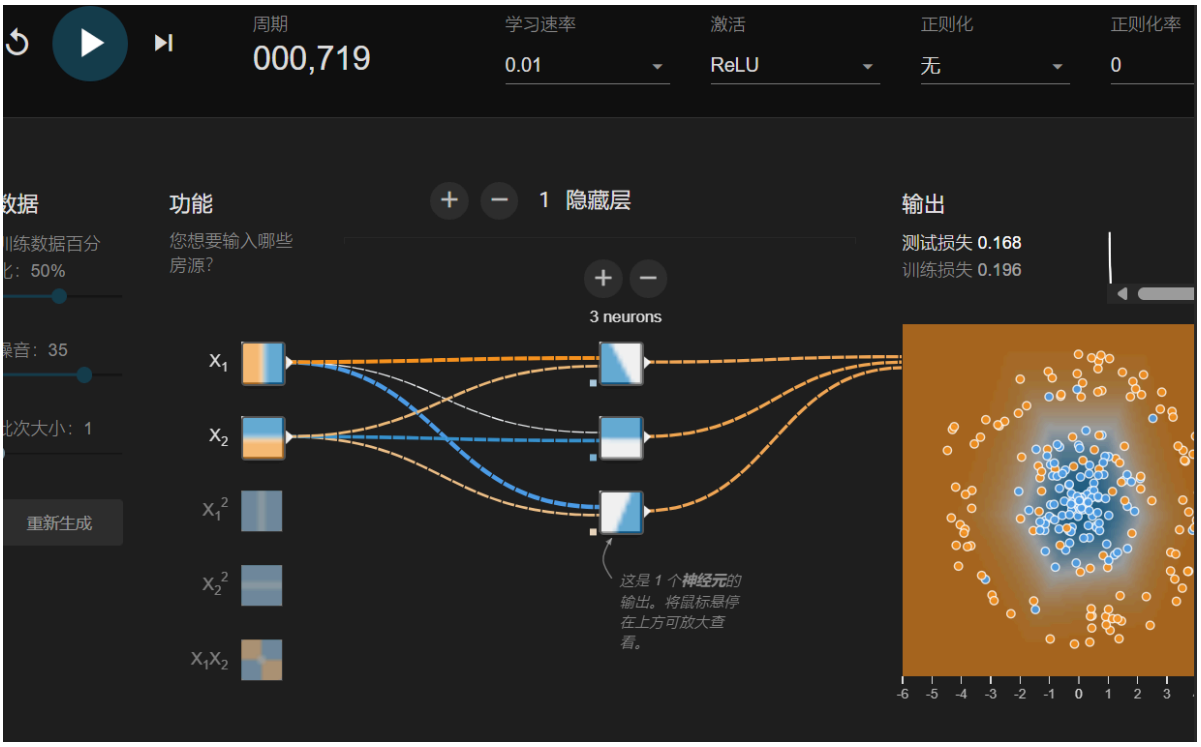
In the interactive widget:

1. Modify the neural network hyperparameters by experimenting with some of the following config settings:
    - Add or remove hidden layers by clicking the **+** and **-** buttons to the left of the **HIDDEN LAYERS** heading in the network diagram.
    - Add or remove neurons from a hidden layer by clicking the **+** and **-** buttons above a hidden-layer column.
    - Change the learning rate by choosing a new value from the **Learning rate** drop-down above the diagram.
    - Change the activation function by choosing a new value from the **Activation** drop-down above the diagram.
2. Click the Play button above the diagram to train the neural network model using the specified parameters.
3. Observe the visualization of the model fitting the data as training progresses, as well as the **Test loss** and **Training loss** values in the **Output** section.
4. If the model does not achieve loss below 0.2 on the test and training data, click reset, and repeat steps 1–3 with a different set of configuration settings. Repeat this process until you achieve the preferred results.

给出满足约束条件的截图，并说明学习到的概念和原理。



一开始调出的最好参数如上图，参考了链接中的解决方案，应该一层就够了。（不过也可以从图中看出，第二层似乎就没什么用）



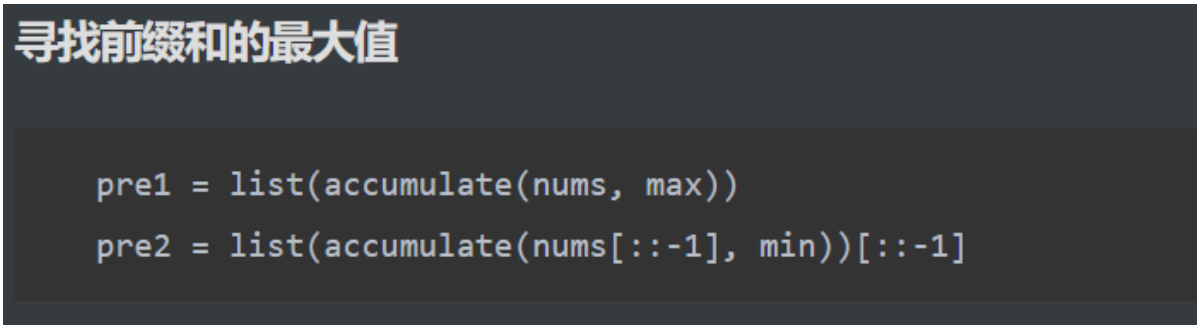
通过图形化的界面大致对神经网络的层数与神经元的作用有了一定的理解，但是感觉学习率的作用还是没有太理解，感觉应该是训练过程中调整参数与训练速度的协调?

2. 学习总结和收获

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。

最近开始跟进leetcode的每日一题，发现了一个小小的优化技巧

2012. 数组美丽值求和 - 力扣（LeetCode）



觉得LC每日一题的难度方差比较大，有的题目还是还有挑战性的。