

# 2021二月月报

## 2021二月月报

### 项目相关

#### 字节跳动前端技术训练营

- 入营过程
- 训练营内容
- 所做工作

#### Nice实习工作

- yolov5实现目标检测
- rotation-yolov5的踩坑记录

#### 算法总结

- 新的数据结构：前缀树Trie
- 新的数据结构：堆（大顶堆）

### 读书摘要

#### 世界简史摘要

#### 经济学原理摘要

### 反思与认识

#### 关于人的灵气

#### 张一鸣：大学四年收获及工作感悟

- 张一鸣在大学里的三点收获：耐心、看书和结交伙伴
- 张一鸣的工作感悟
- 优秀年轻人的五个特点
- 个人感悟

#### 一个中科大程序员的8年工作总结

#### 关于读书

### 发现新大陆

#### 记笔记神器：Typora

#### 《苏幕遮》真好听～

# 项目相关

## 字节跳动前端技术训练营

### - 入营过程

之前参与LeetCode算法周赛，给字节提供了个人邮箱地址，一月初发来邀请函邀请报名前端训练营。自身一直都有字节实习的计划，但没有考虑过前端实习，无奈只有前端的训练营，本着学知识+体验公司文化的态度报了名，参加了机试。

机试五道算法题，在牛客网上作答，限时90分钟。5道题整体难度较低，印象深的考了栈和二叉树，最后一题做过，给出二叉树的前序遍历和中序遍历写出后序遍历，看到题我都乐了，哈哈。这是我第一次真正意义上参加的线上机试，加之不太熟悉牛客网的编译环境，浪费了一点时间，一个小时全部ac提交，第二天收到入营通知。

### - 训练营内容

整个训练营的营员背景还是比较强的，有些出乎意料。入营70多人，C9中的7所有学生参与，同时涵盖了多数985高校，以及包括北邮、西电在内的信息类顶尖211高校，研二学生居多，很多同学都是冲着前端来的，所以有一些基础。

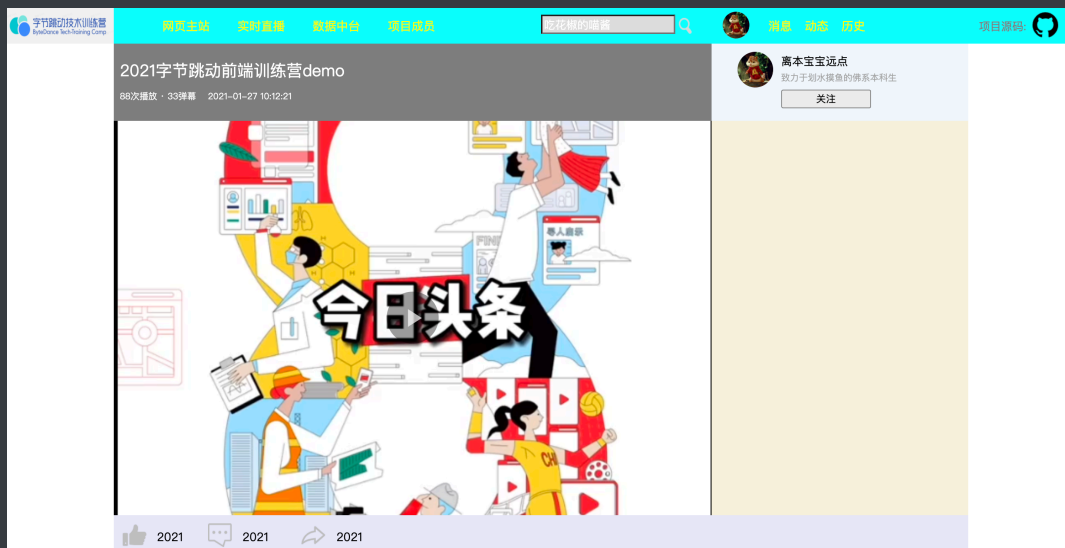
训练营的形式没有我想象中的高大上，最开始就给出大作业，并分好组，然后有4节视频录播课，我只看了两节（还有两节真忘了，还算出勤，不知道会不会有影响），课程结束就每个小组做项目提交就完事了

项目是“我和我的抖音”，要求实现类抖音的短视频App的基础功能，主要包括视频的上下滑动、点赞评论、直播功能、以及数据平台的搭建等等。我们组开始得比较晚，中期验收时都还没起步，进展比较缓慢。

### - 所做工作

#### ■ 前端

一开始是要求用React框架实现前端的，后来也放宽了要求，但我们组最终还是用的React。大家都不太熟悉，进展比较缓慢，我其实一开始做了一个简单的web端页面



后来中期验收老师给了一个demo，大家就照着demo做了。我自己因为前端和后端都懂一点，就自己也实现了一下：



主要借助了西瓜播放器API来实现对视频的控制和播放，用Swiper实现了视频的轮播和鼠标拖动效果，后台基于Flask，做一个数据库用来存储视频数据，前端调用即可。前端的细节渲染其他同学做了，我帮他们把接口部分的问题解决。

总体来说，整个项目还是以前端为主的（主要难点在前端），这个项目也刷新了我对前端的认识，之前觉得能写html，能写js和css就学会前端了，现在看来还差得很远。整个项目完全实现起来还是需要时间的，尤其对于没什么前端经验的同学来说，有非常大的锻炼价值。这个项目即使在结束以后还是非常值得重新完整的实现一次的，能极大提升前端经验。都是后话了。

## ■ 后端

感觉我写后端到了一个瓶颈阶段，各种功能也能大差不差的实现，已经有一段时间没有遇到新的问题了。很多时候这是由于一个接口的实现可以有多种方法，但我一直用我熟悉的方法，而且这个方法往往很不安全，然后我又懒得调整，就导致一直没有提高。

### 1. Flask实现视频属性接口：

```
@app.route('/api/getVideoList')
def get_video_list():
    return jsonify({'data': utils.get_video_list()}) # 这个函数从数据库
拉取视频数据
```

### 2. 从数据库拉取数据代码

```
def get_video_list():
    sql = 'select * from video_info'
    res_list = []
    video_list = query(sql)
    for idx, video in enumerate(video_list):
        video_dic = {
            'Id': video[0],
            'Author': video[1],
            'Description': video[2],
            'TagList': video[3],
            'Likes': video[4],
            'Comments': video[5],
            'Poster': video[6],
            'URL': video[7]
        }
        res_list.append(video_dic)
    return res_list
```

## Nice实习工作

我1月15号从Nice离职，但是还有一些工作没有做完，所以假期在家做了一段时间。目前还没达到预期的要求，但终归出了一个最初的版本。主要工作是做出6个目标检测模型分别检测球鞋的各个角度，以及鞋标OCR检测模型。这一方面可以部署在移动端提高输入数据的质量，一方面可以作为算法的数据预处理部分，提高检测精确度。

## - yolov5实现目标检测

之前目标检测一直用的就是yolov5，所以重新定义了“目标”后做起来也是轻车熟路。Labellmg标记数据，得到yolo专用的label，调整一下文件路径就可以训练。做出6个模型后我也尝试了用单个模型检测全部6个目标，效果也相当可以：



## - rotation-yolov5的踩坑记录

rotation-yolov5是基于yolov5进行的调整版，据我的观察，这个版本并没有改变原先yolov5的模型结构，应该只是增加了一个分类任务，用于旋转检测目标（将角度分成180类）。如果我的观察是正确的话，那相对于原始的yolov5模型，这个模型要想拟合就可能需要大量的数据（毕竟有一个180类的分类任务），而我的数据集并不大，而且要做出6个类的训练数据成本太大了（原始的yolov5每个类50条训练数据跑出来的效果都相当不错）

与此同时，rotation-yolov5的作者并没有利用通用的标记工具来制造label标签（我不知道他具体是怎么做的），所以他基于自己的算法所定义的新的标签有些令人费解，我花了很多功夫在调整标签上（因为一开始我总是觉得一定是我的标签打得有问题才导致模型一直不拟合），可后面不管我怎么调整，跑出来的模型效果就是非常差（几乎没有学习到特征）

当时在尝试与作者进行联系，希望能获得一些帮助，但目前未收到答复。这个模型困扰了我一周的时间，真的是各种尝试，但很遗憾直到最后都没有一个有效的模型可供预测，感觉还是算法本身存在一些问题。我感觉这种项目开源以后还是应该给出一些测试代码的，至少应该让读者能实现一个简单的训练demo，以此了解训练所需要的各项条件，以确保算法的正确性与可扩展性。否则自己盲目地尝试效率真是太低了。

其实有了这个想法以后，是不是可以考虑在他的思路基础上，自己写一个类似的额外的分类模型？

## 算法总结

### – 新的数据结构：前缀树Trie

一种用于检索数据串字符集中的数据结构，广泛用于搜索建议、拼写检查、IP路由及（九宫格）打字预测等涉及字符串预测的领域中。尽管哈希表也可以在 $O(1)$ 时间内完成键值搜索功能，却无法实现：找到具有同一前缀的全部键值，以及按字典序枚举字符串的数据集。

Trie树是一个多叉树，如果只考虑（小写）单词，则每个结点可能包含至多26个分支（a-z）。插入键时可考虑通过循环实现对每一结点的处理，下面基于Python以26个单词的前缀树为例：

```
class Trie:
    def __init__(self):
        self.lookup = {}

    def insert(self, word: str):
        tree = self.lookup
        for a in word:
            if a not in tree:
                tree[a] = {}
            tree = tree[a]
        tree['#'] = '#'

    def search(self, word: str):
        tree = self.lookup
        for a in word:
```



```

        if a not in tree:
            return False
        tree = tree[a]
    return True if '#' in tree else False

def startsWith(self, prefix):
    tree = self.lookup
    for a in prefix:
        if a not in tree:
            return False
        tree = tree[a]
    return True

```

## – 新的数据结构：堆（大顶堆）

堆排序的思想：堆顶的元素是堆中最大元素，将堆顶元素与最后一个元素交换，此时最大元素到了正确位置。然后用一次向下筛选将新的堆顶元素（最大值）移到堆顶，不断遍历直到剩下一个元素

```
'''
```

堆排序（大顶堆）

1. 将数组考虑成一个完全二叉树
2. 先构造堆序
3. 要求二叉树所有的非叶结点必须大于其两个子结点
4. 当满足第二条后将根结点和最后一个结点（此时最后一个结点满足要求不再参与）
5. 重复3, 4

```
'''
```

```
nums = [5, 6, 8, 1, 2, 4, 9]
```

```
def heap_sort():
```

```

    def adjust_heap(idx, length):
        left = idx * 2 + 1
        right = left + 1
        max_loc = idx
        if left < length and nums[left] > nums[max_loc]:

```

```
        max_loc = left
    if right < length and nums[right] > nums[max_loc]:
        max_loc = right
    if max_loc != idx:
        nums[max_loc], nums[idx] = nums[idx], nums[max_loc]
        adjust_heap(max_loc, length)

length = len(nums)
# 构造堆序
for i in range(length//2-1, -1, -1):
    adjust_heap(i, length)

# 交换首尾结点，并通过向下筛选重新构造堆序
for i in range(1, length):
    nums[0], nums[-i] = nums[-i], nums[0]
    adjust_heap(0, length-i)

heap_sort()
print(nums)
```

## 读书摘要

### 世界简史摘要

### 经济学原理摘要

## 反思与认识

### 关于人的灵气

我喜欢看别人的眼睛。直到现在，走在路上我都会感觉到有些迎面走来的人的眼睛里有一种灵气。我也没法定义什么是“灵气”，但是我就是能感觉到，他们的眼睛比常人的更亮，我觉得有灵气的人一定是聪明的。



但大多数时候，我只能在小孩子的眼睛里看到这种灵气。我觉得小时候的我也有灵气，翻我自己小时候的照片我都能隐隐约约感觉得到。那时候的我调皮捣蛋，在幼儿园惹是生非，班上几乎每个小朋友都被我打过...不过现在我觉得这没啥，小朋友就应该活泼好动，大人才考虑该做与不该做，小朋友只管让自己开心～

但很遗憾，随着年龄不断增长，我发现自己身上的灵气慢慢地消失了。原本的活泼也渐渐内敛，取而代之的是沉默少言。我觉得可能人的灵气也会被生活磨平棱角吧，毕竟随着年岁的增长，要考虑的事情越来越多，糟心的事也越来越多，久而久之就产生了一种“麻木”感——哎，这件事要不就这样吧，算了。

我觉得，“灵气”的消失和当下快节奏的生活关系密切，小朋友从上学后就开始不再无忧无虑了，现在的小学生做作业居然都能做到11点我是万万没想到的...另外，互联网技术的不断发展让人的心变浮躁了，青少年热衷于电子游戏，成年人就一整天刷着抖音，一条又一条乐此不疲。久而久之，原本再明亮的眼睛也黯淡无光了...

消失的“灵气”会回来吗？我希望它会回来（当然如果它还残留在我的躯壳内当然更好啦）。参考自身经历和他人的想法，我认为：**尽可能减少电子设备的使用，多看看书，多运动，多去接触和感受自然，闻闻花朵树叶**，把自己短暂地从快节奏的生活中剥离出来，是非常有效的方法，我也应当有意识地付诸实践，找回自己的“灵气”！

## 张一鸣：大学四年收获及工作感悟

### – 张一鸣在大学里的三点收获：耐心、看书和结交伙伴

1. 耐心：能独处，并基于长期思考做判断，而不为短期因素所干扰。如何面对枯燥的生活？阅读大量的人物传记，发现伟大的人在成为伟大的人之前，也是过着枯燥的生活，每天都在做一些微不足道的事情，但这些事情最后从点连成了线，成就了他们。
2. 看书：用别人打游戏、打牌的时间阅读各种各样的书。各种各样的知识连成了线，帮助张一鸣理解行业、理解管理，更快地掌握不熟悉的领域，提高效率。
3. 结交了很多伙伴：并不怎么参与集体活动，交际主要靠修电脑和编程建网站，在这期间结识了女朋友，今天的太太。

### – 张一鸣的工作感悟

1. 责任感：从开始的普通工程师，到第二年负责四五十个人的团队。工作时，不区分哪些工作时该做的，哪些工作是别人的。负责技术，但遇到产品上有问题，也会积极的参与讨论，想产品的方案。**你的责任心，想把事情做好的动力，会驱使你做更多事情，让你得到**

很大的锻炼。当时还是工程师的张一鸣参与的产品经历，对后来的转型有很大帮助。

2. 创业初心：对Elon Mask的崇敬。特别欣赏这种追求卓越和领先的勇气。**年轻人关键是立志高远，享受拼的过程，不自满，不懈怠。要把成功的目标，设定的尽可能远。**

## – 优秀年轻人的五个特点

1. 有好奇心，能够主动学习新事物，新知识和新技能

有一个前同事，基础非常好，但只是把自己的工作做完就下班，对新技术、新工具都不去了解，非常依赖别人。如果是有好奇心的人，前端、后端、算法都去掌握，那么很多调试分析，自己一个人就可以做。

2. 对不确定性保持乐观

很多时候有些事情确实没有把握，但当时就去学，就去看所有相关的东西，最后也许不一定成功，但这个过程会很有帮助——只要对事情的不确定性保持乐观，你会更愿意去尝试。

3. 不甘于平庸

如果一个人一毕业就把目标定在北京买一套房子，把精力都花在这上面，那么工作就会受到影响，他的行为会发生变化，不愿意冒风险。如果不甘于平庸（这里指对自己的标准一定要高），就不会为这些事情操心，也许前两年变化的慢，但10年以后再看，会非常不一样。

4. **不傲娇，要能延迟满足感**

两个印象比较深刻的年轻人，素质、技术都不错。他们觉得其他同事做的不如他们，其实不是：**他们确实可以算作当时同时里面的TOP20%，但他们都认为自己是TOP1%**。所以很多基础一点的工作，就不愿意做，或者需要跟同事配合的工作，就配合的不好。本来都是资质非常好的人才，人非常聪明、动手能力也强，但没有控制好自己的傲娇情绪。**不甘于平庸**是你的目标要设得很高，**不傲娇**是你对自己现状要踏实。

另一个例子是一个做产品的同事，当时大家都觉得他不算特别聪明，就让他做辅助性的工作，但现在他已经是一家十亿美金公司的副总裁。它的特点就是肯去做，负责任，从不推诿，只要有机会承担的事情他总可以尽可能的做好。每次也不算做的特别好，但我们总能给他反馈。

5. 对重要的选择有判断力

选什么专业、选什么公司、选什么职业、选什么发展路径，自己要有判断力，不要被短期选择所左右。原先有很多人愿意去外企，不愿意去新兴公司，但实际上，能摆脱从众心理，有自己判断的人，不是特别多。

## - 个人感悟

- 一个人的成功真的不是没有原因的，张一鸣确实对事物看得比常人通透。
- 针对提到的优秀年轻人的特征，要学习。多看书，这确实能潜移默化的影响一个人的方方面面，至少不至于让自己写个周报都感到别扭。
- 对于提到的年轻人身上存在的问题，有则改之，无则加勉。尤其是不傲娇，今后一定要注意这个问题。不论对待什么工作都不能懈怠；对待布置的任务要有责任感，有些事情尽管不是自己的范畴，也应该尽力协助；与同事合作态度要积极正面，一定不能因为自己提前掌握了一些知识而轻视别人。

## 一个中科大程序员的8年工作总结

- 从2013年毕业后加入上海航天X院，再到入职华为，从华为离职
- 如果你的亲人是普通阶层，那对于人生中一些大事来说，他们给的建议往往就是普通阶层的思维，他们的阶层就是他们一生思维决策的结果，如果你的目标是跳出本阶层，那最好只把他们的建议当成参考
- 和你的同事比工资没有意义，比工资总会有人受伤，更多的是负面影响，并且很多时候受伤的是你
- 不要说一项技能没有用，任何你掌握的技能都有价值，但是你要学会找到发挥它的场景。工作中如果要上升，你要勇于承担麻烦的、有挑战的任务，当你推掉麻烦的时候，你也推掉了机遇
- 很多人喜欢不停的工作，但不会停下来思考，缺乏总结与复盘的能力，其实阶段性总结复盘，不仅能够固化前面的经验，也能梳理后面的方向；把事情做对很重要，但是更重要的是做对的事；方向正确慢就是快
- 不要因为某人的外在，如外貌、习惯、学历等对人贴上标签，去盲目否定别人，对于别人的建议，应该从客观出发，综合分析，从善如流是一项非常难得的品质（从善如流：能迅速而顺畅地接受别人的意见）
- 人很难挣到他认知之外的财富，就算偶然拿到了，也可能很快失去。所以不要羡慕别人投机获得的财富，努力提升你的思维，财商才是正道
- 建议养成写技术总结文章的习惯，这不仅能提升个人技术，同时分享也可以增加你的影响力

## 关于读书

- 腹有诗书气自华，古人诚不欺我...肚子里没墨水，真的绞尽脑汁都写不出让自己能满意的文字
- 多掌握一些自己领域以外的其他知识有时候也有巨大帮助
- 读书可以让自己静下来，让自己摆脱浮躁的情绪，暂时地脱离快节奏的生活，陶冶情操～

## 发现新大陆

### 记笔记神器：Typora

大一用笔记本记笔记，数学笔记做了一整本，密密麻麻全是题；大二用OneNote记笔记，当时感觉没有什么比这个更好用了；大三发现了Xmind，又换成了思维导图笔记；大四换了电脑，原来的OneNote在Mac环境下简直让人抓狂，Xmind也没有更多亮点，且不利于导出，于是一直在探索新的方式来记笔记。有段时间学会了LaTeX，用LaTeX做笔记，后来因为效率太低，也放弃了。直到近期发现Markdown作出的笔记非常清晰，就考虑用Markdown作为主要的记笔记工具。

PyCharm中自带Markdown的插件，可以使用，但是不能导出，还是很麻烦。最终发现了Typora，简直是记笔记超级神器。轻量级编辑器，解压后不过20Mb，简洁的页面，几乎跟文本编辑器一模一样，支持各种编程语言，支持目录生成，支持导出，各种背景供选择，直接起飞！

《苏幕遮》真好听～