

faiss 没有提供余弦距离怎么办

参考:<https://zhuanlan.zhihu.com/p/40236865>

faiss是Facebook开源的用于快速计算海量向量距离的库,但是没有提供余弦距离,而余弦距离的使用率还是很高的,那怎么解决呢

答案说在前面

```
knowledge_embedding = np.random.random((1000, 300)).astype('float32') # 1000个待查知识点
query_embedding = np.random.random((100, 300)).astype('float32') # 100个查询语句
normalize_L2(knowledge_embedding) # 熟悉余弦相似度公式的都知道,点击后会除于长度,所以要把长度归一化到1,就可以直接点击算出余弦相似度
normalize_L2(query_embedding) # 熟悉余弦相似度公式的都知道,点击后会除于长度,所以要把长度归一化到1,就可以直接点击算出余弦相似度
index = faiss.IndexFlat(d, faiss.METRIC_INNER_PRODUCT) # 等价 index=faiss.IndexFlatIP(d)
index.add(knowledge_embedding) # 把知识点加到索引里面

D, I =index.search(query_embedding, k=5) # 召回5个
```

进一步实验

```
import faiss
from faiss import normalize_L2
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import copy

def faiss_cos_similar_search(x, k=None):
    #
    assert len(x.shape) == 2, "仅支持2轴向量的距离计算"
    x = copy.deepcopy(x)
    nb, d = x.shape
    x = x.astype('float32')
    k_search = k if k else nb
    normalize_L2(x)
    index = faiss.IndexFlat(d, faiss.METRIC_INNER_PRODUCT)
    # index=faiss.IndexFlatIP(d)
    # index.train(x)
    # index=faiss.IndexFlatL2(d)

    index.add(x)
    D, I =index.search(x, k=k_search)
    return I

def sklearn_cos_search(x, k=None):
    assert len(x.shape) == 2, "仅支持2轴向量的距离计算"
    x = copy.deepcopy(x)
    nb, d = x.shape
    ag=cosine_similarity(x)
    np.argsort(-ag, axis=1)
    k_search = k if k else nb

    return np.argsort(-ag, axis=1)[: , :k_search]
```

公告

昵称: littlepai
园龄: 4年10个月
粉丝: 12
关注: 2
+加关注



2022年2月						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5
6	7	8	9	10	11	12

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

距离(2)
向量(2)
faiss(2)
知乎爬虫(2)
自动验证码识别(2)
迁移学习(2)

```
def test_IndexFlatIP_only(nb = 1000, d = 100, kr = 0.005, n_times=10):
    k = int(nb * kr)
    print("recall count is %d" % (k))
    for i in range(n_times):

        x = np.random.random((nb, d)).astype('float32')
        # x = np.random.randint(0,2, (nb,d))
        # faiss_I = faiss_cos_similar_search(x, k)
        index=faiss.IndexFlatIP(d)
        index.train(x)
        index.add(x)
        D, faiss_I =index.search(x, k=k)

        sklearn_I = sklearn_cos_search(x, k)

        cmp_result = faiss_I == sklearn_I

        print("is all correct: %s, correct batch rate: %d/%d, correct sample rate: %d/%d" % \
              (np.all(cmp_result), \
               np.all(cmp_result, axis=1).sum(), cmp_result.shape[0], \
               cmp_result.sum(), cmp_result.shape[0]*cmp_result.shape[1] ) )

def test_embedding(nb = 1000, d = 100, kr = 0.005, n_times=10):
    k = int(nb * kr)
    print("recall count is %d" % (k))
    for i in range(n_times):

        x = np.random.random((nb, d)).astype('float32')
        # x = np.random.randint(0,2, (nb,d))
        faiss_I = faiss_cos_similar_search(x, k)
        sklearn_I = sklearn_cos_search(x, k)

        cmp_result = faiss_I == sklearn_I

        print("is all correct: %s, correct batch rate: %d/%d, correct sample rate: %d/%d" % \
              (np.all(cmp_result), \
               np.all(cmp_result, axis=1).sum(), cmp_result.shape[0], \
               cmp_result.sum(), cmp_result.shape[0]*cmp_result.shape[1] ) )

def test_one_hot(nb = 1000, d = 100, kr = 0.005, n_times=10):
    k = int(nb * kr)
    print("recall count is %d" % (k))
    for i in range(n_times):

        # x = np.random.random((nb, d)).astype('float32')
        x = np.random.randint(0,2, (nb,d))
        faiss_I = faiss_cos_similar_search(x, k)
        sklearn_I = sklearn_cos_search(x, k)

        cmp_result = faiss_I == sklearn_I

        print("is all correct: %s, correct batch rate: %d/%d, correct sample rate: %d/%d" % \
              (np.all(cmp_result), \
               np.all(cmp_result, axis=1).sum(), cmp_result.shape[0], \
               cmp_result.sum(), cmp_result.shape[0]*cmp_result.shape[1] ) )

if __name__ == "__main__":

    print("test use IndexFlatIP only")
    test_IndexFlatIP_only()
    print("-"*100 + "\n\n")
    print("test when one hot")
    test_one_hot()
    print("-"*100 + "\n\n")
    print("test use normalize_L2 + IndexFlatIP")
    test_embedding()
    print("-"*100 + "\n\n")
```

tensorflow(2)

批标准化(1)

层标准化(1)

余弦(1)

更多

随笔分类

DB(2)

ORACLE(2)

PYTHON(4)

机器学习(3)

开发(3)

深度学习(6)

随笔档案

2020年8月(1)

2020年4月(1)

2019年12月(2)

2018年12月(1)

2018年2月(2)

2017年12月(1)

2017年11月(3)

2017年5月(1)

2017年4月(1)

文章分类

DB(1)

PYTHON(1)

相册

流程图(6)

磨刀石

Cmd Markdown 公示指导手册

阅读排行榜

1. 最小二乘法-公式推导(42381)
2. 深度学习与爬虫实例教学--深度学习模型构建和训练(3377)
3. 深度学习与爬虫实例教学--项目基本介绍和体验(2424)
4. faiss 没有提供余弦距离怎么办(2320)
5. 正负样本比率失衡SMOTE(2112)

评论排行榜

1. 最小二乘法-公式推导(3)
2. 深度学习与爬虫实例教学--深度学习模型构建和训练(2)

推荐排行榜

1. 最小二乘法-公式推导(5)
2. 深度学习与爬虫实例教学--深度学习模型构建和训练(2)
3. 深度学习与爬虫实例教学--项目基本介绍和体验(1)

下面是实验结果, 比较faiss和sklearn实现的余弦相似度召回顺序是不是完全一样

```
## -- End pasted text --
test use IndexFlatIP only
recall count is 5
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1255/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1299/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1196/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1231/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1266/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1257/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1303/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1283/5000
is all correct: False, correct batch rate: 0/1000, correct sample rate: 1307/5000
is all correct: False, correct batch rate: 1/1000, correct sample rate: 1269/5000
```

```
test when one hot
recall count is 5
is all correct: False, correct batch rate: 831/1000, correct sample rate: 4721/5000
is all correct: False, correct batch rate: 848/1000, correct sample rate: 4744/5000
is all correct: False, correct batch rate: 823/1000, correct sample rate: 4719/5000
is all correct: False, correct batch rate: 857/1000, correct sample rate: 4762/5000
is all correct: False, correct batch rate: 849/1000, correct sample rate: 4752/5000
is all correct: False, correct batch rate: 854/1000, correct sample rate: 4751/5000
is all correct: False, correct batch rate: 844/1000, correct sample rate: 4741/5000
is all correct: False, correct batch rate: 840/1000, correct sample rate: 4745/5000
is all correct: False, correct batch rate: 856/1000, correct sample rate: 4753/5000
is all correct: False, correct batch rate: 820/1000, correct sample rate: 4689/5000
```

```
test use normalize_L2 + IndexFlatIP
recall count is 5
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
is all correct: True, correct batch rate: 1000/1000, correct sample rate: 5000/5000
```

分析:第一份结果(横线隔开), 是仅用IndexFlatIP的时候, 跟余弦距离的结果相差非常大

第二份结果, 是当数据是 one hot 的时候, 用 normalize_L2 + IndexFlatIP, faiss和sklearn召回不完全一样是因为余弦相似度相同的时候召回id排序不同而已

第二份结果, 是当数据是 embedding 的向量的时候, 用 normalize_L2 + IndexFlatIP, faiss和sklearn召回一般都会全部对得上, 因为相同距离的情况很少会出现

分类: [深度学习](#), [机器学习](#), [开发](#)

标签: [faiss](#), [cos](#), [余弦](#), [距离](#), [向量](#)

[好文要顶](#)[关注我](#)[收藏该文](#)



littlepai

关注 - 2

粉丝 - 12

[±加关注](#)

« 上一篇: [faiss 占用cpu过高](#)

» 下一篇: [matplotlib 中文乱码](#)

0

 推荐

0

 反对

4. 重建主键索引为非压缩索引(1)

5. ORACLE聚合函数细节(1)

最新评论

1. Re: Python将数据库的父子关系表画成树形结构

大佬, 这代码有错呀, 循环运行不了, 我不知道怎么改好, 请教下 Traceback (most recent call last):
File "C:/Users/chaoqun/OneDrive/桌面/...

--天上昭

2. Re: 最小二乘法-公式推导

谢谢分享, 学习了

--wdliming

3. Re: 最小二乘法-公式推导

看懂了, 感谢分享。另外, 第四步式子后面少了一个平方

--蜗牛JC

4. Re: 最小二乘法-公式推导

十分感谢, 学习了!!

--schwarzeni00


5. Re: 深度学习与爬虫实例教学--深度学习模型构建和训练

@ 愤怒的TryCatch共同学习, 这个主要是能识别真实生产环境的验证码, 不是自己生成自己识别那种, 所以我觉得有必要看一下我这个教程的, 帮我fork一下github, 共同加油...

--littlepai

posted @ 2019-12-31 12:47 littlepai 阅读(2320) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论. 立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

编辑推荐:

- 2021 .NET Conf China 主题分享之-轻松玩转.NET大规模版本升级
- 理解 OAuth2.0 协议和授权机制
- Asp.net core IdentityServer4 与传统基于角色的权限系统的集成
- 记一次 .NET 某供应链WEB网站 CPU 爆高事故分析
- 从 Mongo 到 ClickHouse 我到底经历了什么？

最新新闻:

- 翻遍整个乌克兰, 竟找不出一个扛把子App
- 俄乌冲突影响太大 iPhone价格大涨: 涨幅达1800多元
- 特斯拉开始在加拿大推出全自动驾驶 FSD Beta 测试版
- 华为应用市场全球月活跃用户达5.8亿 未来5年到10年聚焦全场景智慧生态
- 苹果第八大股东表态: 将投票反对苹果管理层薪酬计划
- » 更多新闻...