Faiss应用

漫漫成长
算法工程师

Faiss是为稠密向量提供高效相似度搜索的框架(Facebook AI Research），选择索引方式是faiss的核心内容，faiss 三个最常用的索引是：IndexFlatL2, IndexIVFFlat,IndexIVFPQ。

1. IndexFlatL2/ IndexFlatIP为最基础的精确查找。应用样例：

```
user_vector_arr  # shape(526,066, 128)
gds_vector_arr   # shape(5,172, 128)
dim = 128# 向量维度
k = 10  # 定义召回向量个数
index = faiss.IndexFlatL2(dim)  # L2距离，即欧式距离(越小越好)
# index=faiss.IndexFlatIP(dim) # 点乘，归一化的向量点乘即cosine相似度(越大越好)
index.add(gds_vector_arr) # 添加训练时的样本
D, I = index.search(user_vector_arr, k) # 寻找相似向量， I表示相似用户ID矩阵， D表示距离矩阵
```

2. IndexIVFFlat称为倒排文件索引，是使用K-means建立聚类中心，通过查询最近的聚类中心，比较聚类中的所有向量得到相似的向量，是一种加速搜索方法的索引。应用样例：

```
user_vector_arr  # shape(526,066, 128)
gds_vector_arr   # shape(5,172, 128)
dim = 128 # 向量维度
k = 10  # 定义召回向量个数
nlist = 100 #聚类中心的个数
quantizer = faiss.IndexFlatL2(dim)  # 定义量化器
index = faiss.IndexIVFFlat(quantizer, dim, nlist, faiss.METRIC_L2) #也可采用向量内积
index.nprobe = 10 #查找聚类中心的个数，默认为1个，若nprobe=nlist则等同于精确查找
index.train(gds_vector_arr) #需要训练
index.add(gds_vector_arr) # 添加训练时的样本
D, I = index.search(user_vector_arr, k) # 寻找相似向量， I表示相似用户ID矩阵， D表示距离矩阵
```

3. IndexIVFPQ是一种减少内存的索引方式，IndexFlatL2和IndexIVFFlat都会全量存储所有的向量在内存中，面对大数据量，faiss提供一种基于Product Quantizer(乘积量化)的压缩算法编码向量到指定字节数来减少内存占用。但这种情况下，存储的向量是压缩过的，所以查询的距离也是近似的。应用样例：

```
user_vector_arr  # shape(526,066, 128)
gds_vector_arr   # shape(5,172, 128)
dim = 128 # 向量维度
k = 10  # 定义召回向量个数
nlist = 100  #聚类中心的个数
m = 8  # 压缩成8bits
quantizer = faiss.IndexFlatL2(dim) # 定义量化器
index = faiss.IndexIVFPQ(quantizer, dim, nlist, m, 8)  # 8 specifies that each sub-ve
index.nprobe = 10 #查找聚类中心的个数，默认为1个，若nprobe=nlist则等同于精确查找
index.train(gds_vector_arr) #需要训练
index.add(gds_vector_arr) # 添加训练时的样本
D, I = index.search(user_vector_arr, k) # 寻找相似向量， I表示相似用户ID矩阵， D表示距离矩阵
```

4. index_factory是faiss实现的一个索引工厂模式,可以通过字符串来灵活的创建索引；PCA算法可将向量降到指定的维度。应用样例：

```
user_vector_arr  # shape(526,066, 128)
gds_vector_arr   # shape(5,172, 128)
dim = 128 # 向量维度
k = 10  # 定义召回向量个数
index = faiss.index_factory(dim, “PCAR32,IVF100,SQ8”)# PCA降到32位;搜索空间100;SQ8,scal
index.train(gds_vector_arr) #需要训练
index.add(gds_vector_arr) # 添加训练时的样本
D, I = index.search(user_vector_arr, k) # 寻找相似向量， I表示相似用户ID矩阵， D表示距离矩阵


# 索引简写可查询:https://github.com/facebookresearch/faiss/wiki/Faiss-indexes
```

=>一般选用欧式距离，速度较快(KNN算法-KDTree? 处理高维向量 - BallTree?)；也可得到距离矩阵后转化为cosine(向量须为L2归一化后), 方便有用户商品对截断需求的阈值确定。

```
cosine = (2 - L2_Distance)/2
```

发布于 2020-10-29 17:24

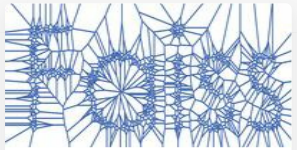
相似度计算

推荐阅读

FAISS 用法

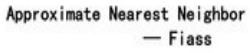
坑挖太多了，已经没时间填。那就再挖一个吧 faiss是为稠密向量提供高效相似度搜索和聚类的框架。由Facebook AI Research研发。 具有以下特性。1、提供多种检索方法 2、速度快3、可存在内存...

汤go



搜索召回 | Facebook: 亿级向量相似度检索库Faiss原理...

魔法学院的... 发表于信息检索



Fiass - 常见问题总结

一小撮人

文本相似度算法WMD(Word Mover's Distance)论文阅...

1. 简介文本间的距离在文本分类, QA问答, 信息检索方面有比较广泛的应用. 最近完成的一个项目就是用bert 向量化文本然后计算余弦距离来实现问答匹配, 这种方法简单粗暴, 但比较依赖特征提取...

warri... 发表于NLP论文...

1 条评论

切换为时间排序

写下你的评论...

知乎用户 2020-12-16

是不是必须把图像转成一维数组，才能使用faiss

赞

▲ 赞同 2 ▼ 1 条评论 分享 喜欢 收藏 申请转载 ···