

# python 实验报告

## 实验名称：PCA

院系：人工智能学院

姓名：张运吉

学号：211300063

班级：21 级人工智能学院 AI2 班

邮箱：211300063@smail.nju.edu.cn

时间：2022 年 5 月 17 日

# 目录

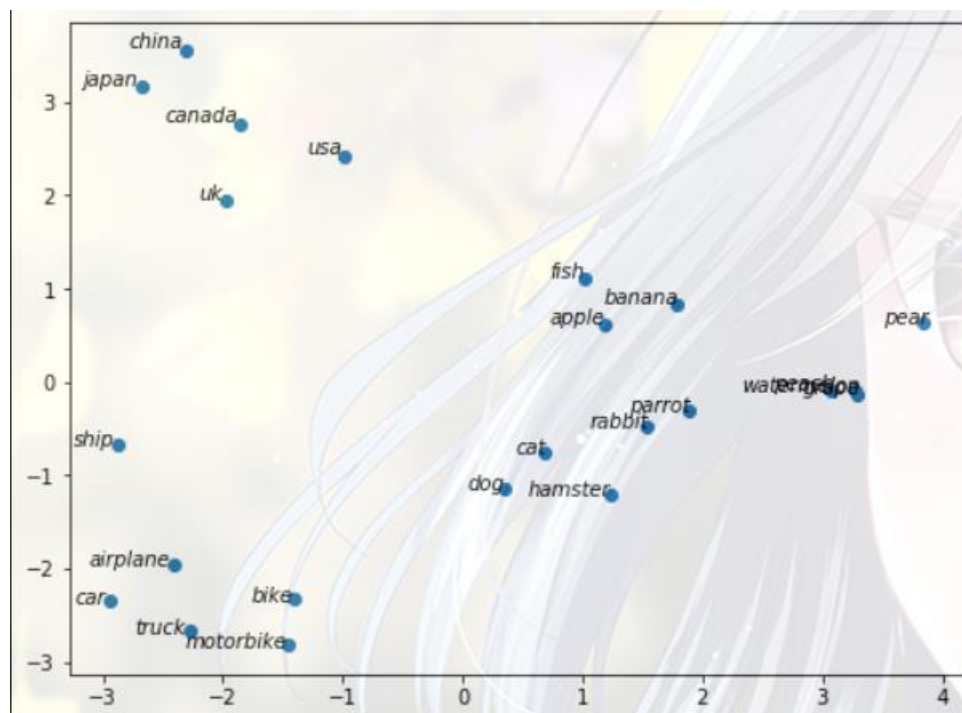
|                         |   |
|-------------------------|---|
| 1.手动实现: .....           | 3 |
| 2 使用 sklearn 库实现: ..... | 4 |
| 3 实验结果: .....           | 4 |

## 1.手动实现：

我们有  $m$  个  $n$  维的样本数据.

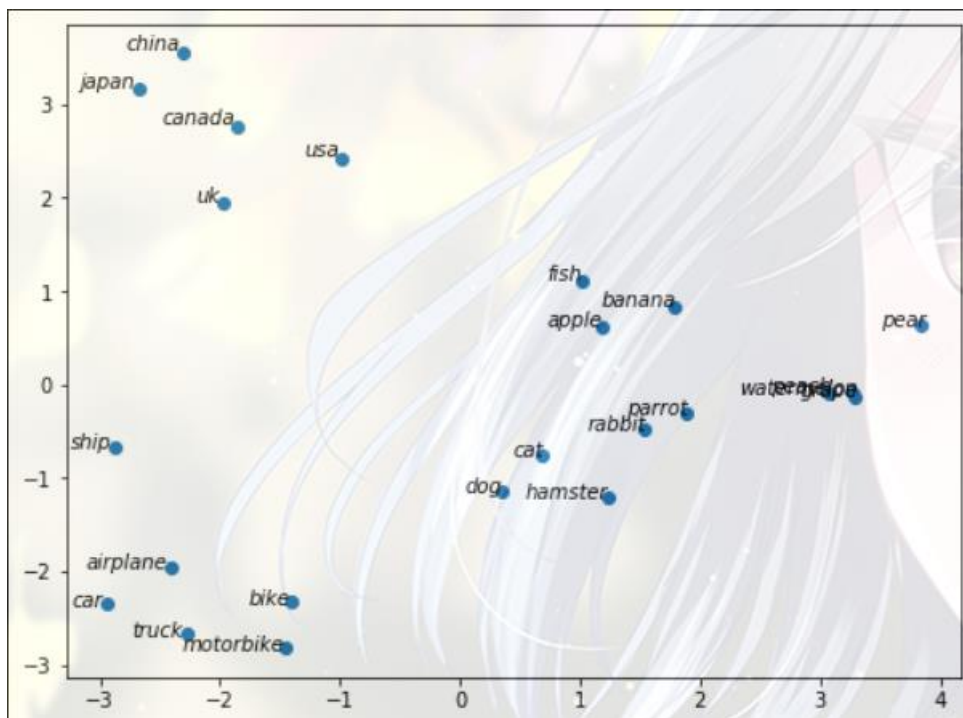
1. 将原始数据转化成为一个  $n$  行  $m$  列的数据矩阵  $X$ ;
2. 每一行减去均值, 进行"中心化";
3. 求出协方差矩阵  $P = \frac{1}{m} X X^T$ ;
4. 求出协方差矩阵的特征值和特征向量, 形成特征矩阵;
5. 计算  $X' = E_k^T X$ ,  $X'$  即为降维过后的数据;
6. 将数据按特征值由大到小排列, 并选取前  $k$  维数据输出.

```
# 使用numpy实现PCA
def pca(data, n_component):
    data = data.T
    data -= data.mean(axis = 1).reshape(-1, 1)
    cov_matrix = 1 / len(data[0]) * data @ data.T
    eigenvalues, eigenmatrix = np.linalg.eig(cov_matrix)
    new_data = eigenmatrix.T @ data
    idx = np.argsort(-eigenvalues)[0:n_component]
    return new_data[idx]
```



## 2 使用 sklearn 库实现：

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
new_data1 = pca.fit_transform(datas_np.T).T
# Draw scatter plot
x = new_data1[0]
y = new_data1[1]
fig=plt.figure(figsize=(8,6))
ax=plt.subplot(1,1,1)
ax.scatter(x, y)
for i in range(len(x)):
    ax.text(x[i]*1.01, y[i]*1.01, words[i], fontsize=10, style =
"italic",horizontalalignment='right', weight = "light", rotation=0) #给散点加标签
plt.show()
```



## 3 实验结果：

两种算法实现的效果相差不大，而且降维后的结果符合预期。