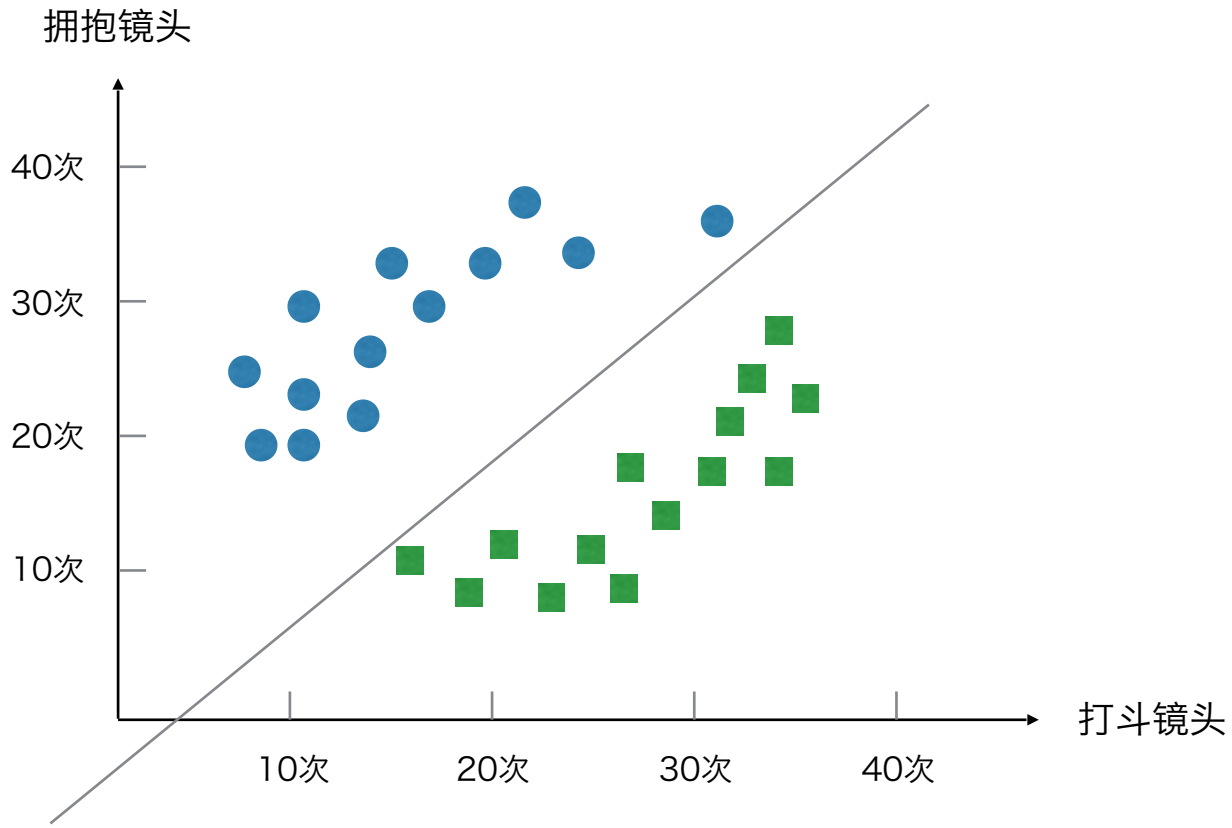
人工智能系列课程

自然语言处理

目录

回顾

线性分类



机器如何画一条直线

直线函数： y=wx+b （w调整斜度,b调整左右移动）

1，先假设w=1，b=1 （设定初始权重值）

2，将每一个点的x带入直线，求y，判断误差（损失函数）

3，修正权重，迭代过程，一点一点微调 （迭代次数，学习率）

4，保存权重和常数，用于预测新数据 （保存模型）

注意点：

修正权重时需要所有点同时计算、w和b

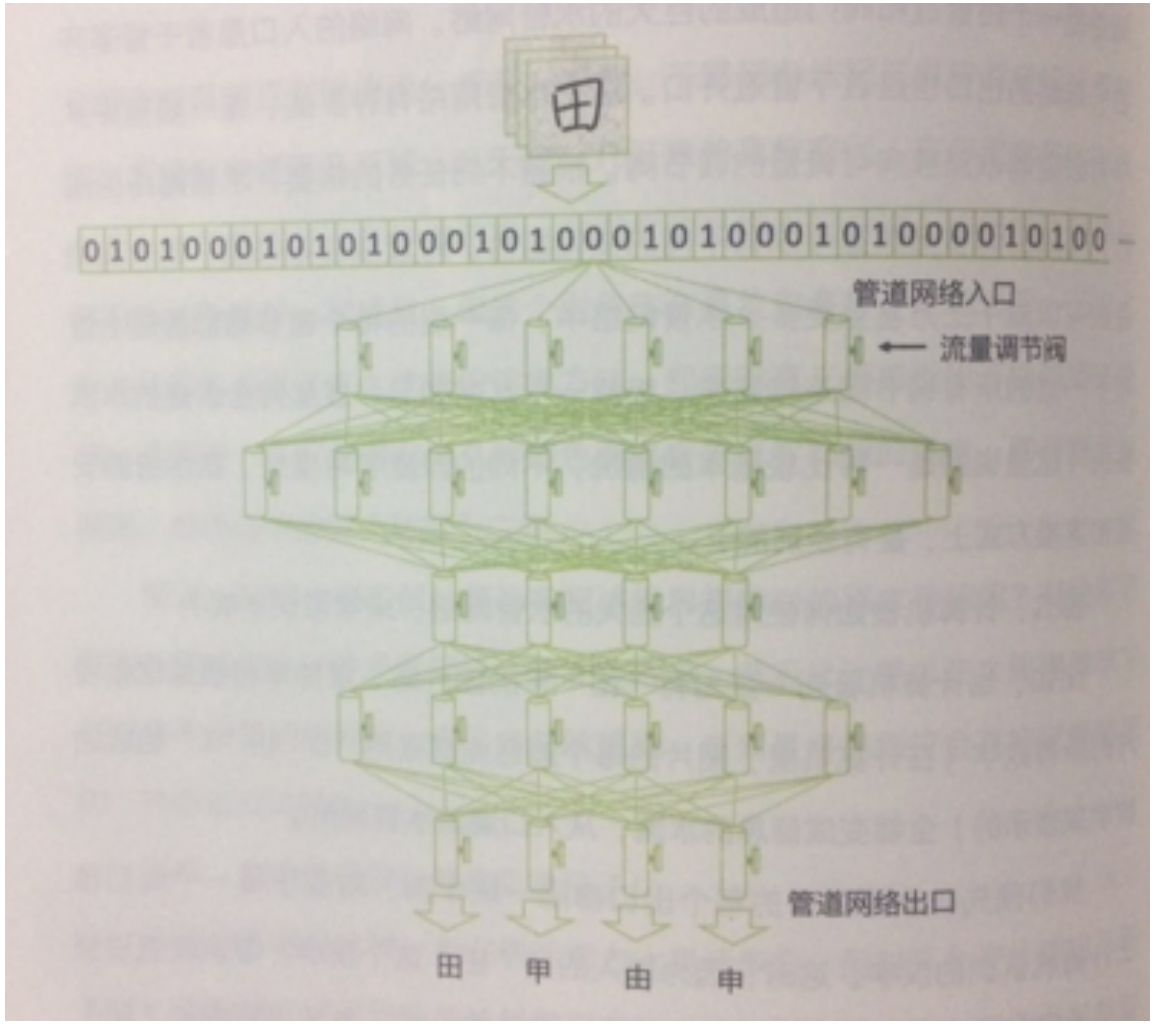
同时计算，如果是多维特征会有多个w

$y=w_1x_1 + w_2x_2+w_3x_3 +b$

线性分类算法： 逻辑回归，LR（Logistic Regression）

回顾

深度学习的原理

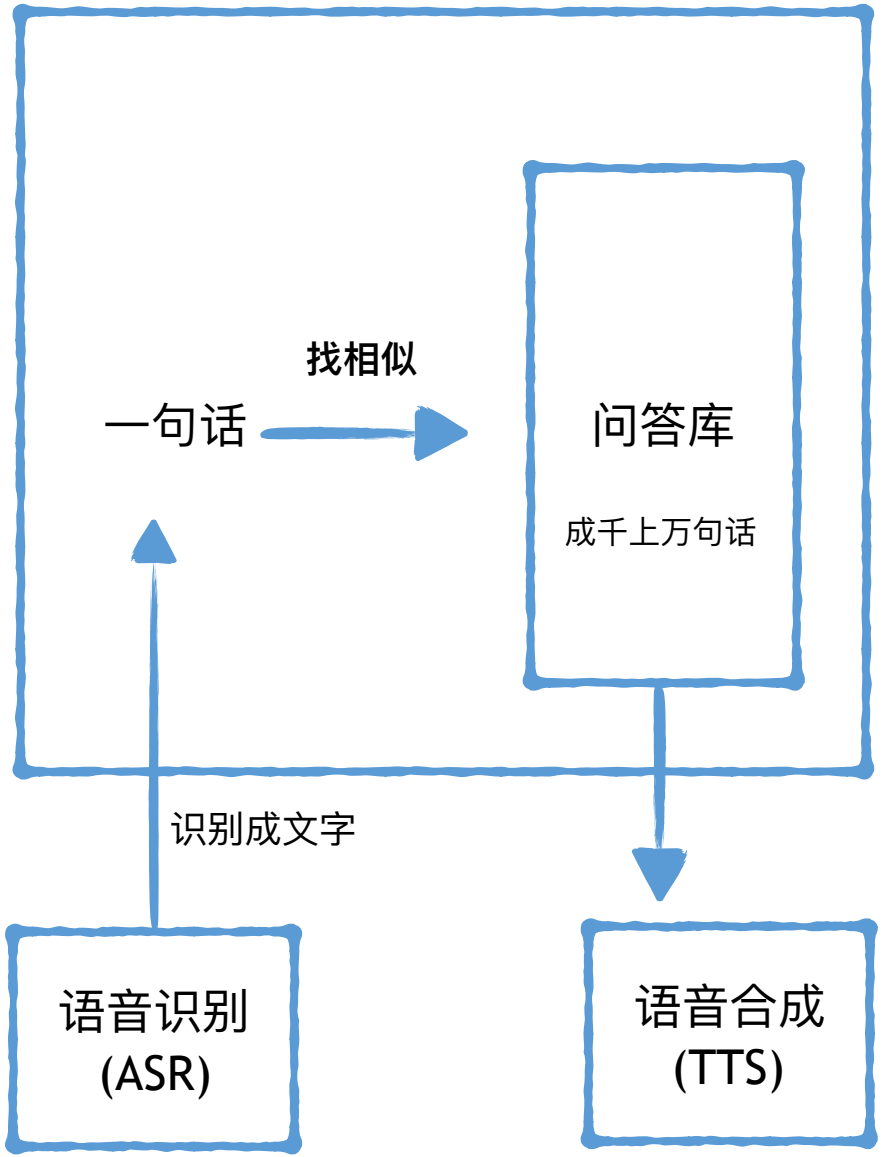


http://playground.tensorflow.org/

自然语言处理的应用


小墨小墨，今天天气怎么样
小墨小墨，我要看动画片
小墨小墨，今天听你讲故事
小墨小墨，我要听电台
小墨小墨，提醒我八点开始写作业

原理



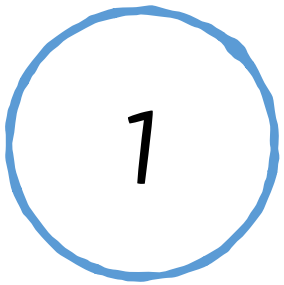
本课案例

```
14785	英国普洱茶饼可以随身带过安检吗?
14786	英国哪裡可以用銀聯卡取英鎊
14787	换护照以后旧护照上的多次签证还能用吗?
14788	请问罗马机场到市区远不远?
14789	意大利米兰火车站叫什么
14790	伦敦的大超市晚上几点关门?
14791	伦敦工厂店地址
14792	伦敦工厂店营业时间
14793	英国普洱茶饼可以随身带过安检吗?
14794	伦敦工厂店怎么样，有什么优惠吗
14795	请问伦敦的大超市晚上几点关门?
14796	卡塔尔航空的飞机 行李限额是30KG么
14797	英国伦敦的特拉法加广场怎么去?
14798	英国family&friends rail card卡可以在火车站办吗
14799	英国牛津街和哈罗德哪个更值得去呢
14800	英国伦敦哈罗德值得去吗
14801	伦敦逛吃逛吃，除了牛津街，还有什么推荐
14802	怎么从伦敦希思罗机场去伯明翰大学！怎么转车?
14803	英国牡蛎卡是哪里买的
14804	有了英国签证，如何去法国？要什么手续吗?
14805	办理法国的签证需要多久?
14806	英国火车上能吃小零食吗?
14807	英国卷发棒能带吗
14808	伦敦工厂店怎么样，有什么优惠吗
14809	怎么去伦敦工厂店
14810	英国比斯特购物村地址
14811	英国的比斯特购物村简介
14812	希斯罗机场每个航站楼都有退税点
```

```
➜  dingdian python tfidf_question.py
请输入一个问题：越南旅游需要带插头转换器吗
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/y5/tjc0ty0x3hv3_pvx741q
Loading model cost 1.633 seconds.
Prefix dict has been built succesfully.
插头要转换器吗？

越南需要带电源转换器吗？插头是和中国一样吗

牙压需要带转换器吗

越南需要带插座转换器吗？

越南需要带插座转换器吗？

越南需要带转换插头吗

越南需要带转换插头吗

越南需要带转换插头吗

需要带转换器吗
```

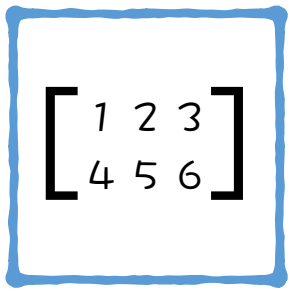
自然语言的向量表示

基本概念



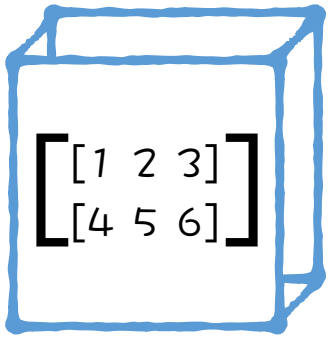
标量

(Scalar)



向量

(Vector)



张量

(Tensor)

语料

人民日报

如何将语料中的词用向量表示?

OneHot编码

语料：

西雅图的博物馆有哪些
住西雅图机场附近酒店有哪些

分词 →

西雅图/的/博物馆/有/哪些
住/西雅图/机场/附近/酒店/有/哪些

西雅图	[1,0,0,0,0,0,0,0,0]
的	[0,1,0,0,0,0,0,0,0]
博物馆	[0,0,1,0,0,0,0,0,0]
有	[0,0,0,1,0,0,0,0,0]
哪些	[0,0,0,0,1,0,0,0,0]
住	[0,0,0,0,0,1,0,0,0]
机场	[0,0,0,0,0,0,1,0,0]
附近	[0,0,0,0,0,0,0,1,0]
酒店	[0,0,0,0,0,0,0,0,1]

表示句子

西雅图的博物馆有哪些
[1,1,1,1,1,0,0,0,0]

住西雅图机场附近酒店有哪些
[1,0,0,1,1,1,1,1,1]

西雅图西雅图
[2,0,0,0,0,0,0,0,0]

机场的博物馆有哪些
[0,1,1,1,1,0,1,0,0]

这种表示方法叫：Bag-of-words （**BOW**） **词袋**

简化表达式：西雅图西雅图博物馆 —> [(1,2),(3,1)]

BOW词袋模型

关于分词

明天台南县的天气

张三说的确实在理

动态规划算法　　HMM模型　　Viterbi算法

Jieba分词

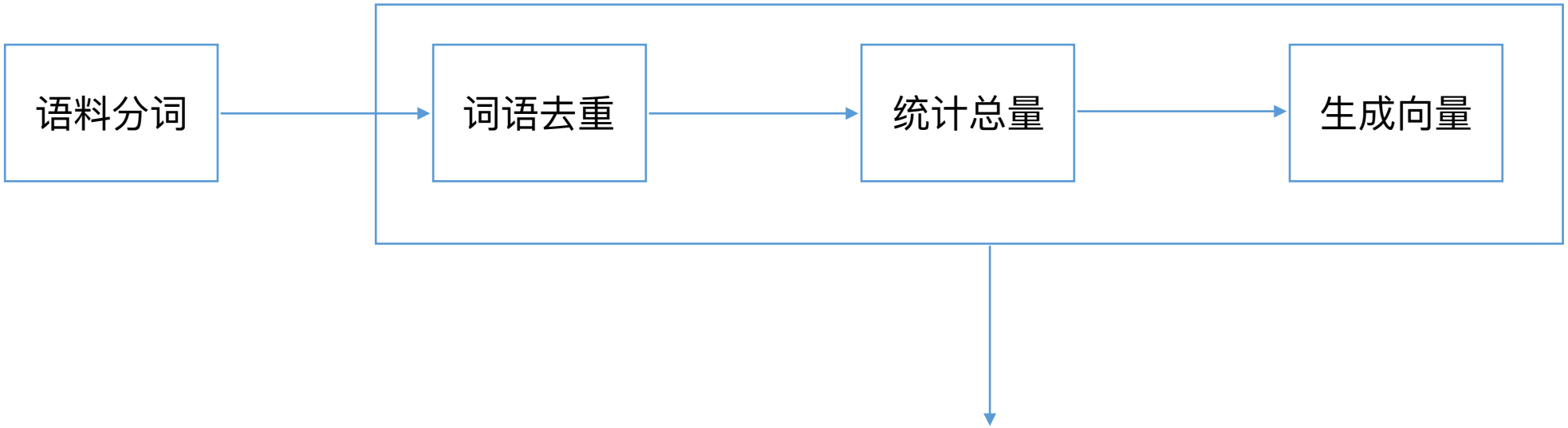
JieBa分词的用法

安装： pip install jieba

jieba.cut(“传一句话”) 进行分词

jieba.add(“词语”) 添加一个词

生成向量



gensim.corpora.Dictionary(sentences)

sentences 为分好词的语料， 格式是二维数组：

```
[
    ["西雅图","的","博物馆","有","哪些"],
    ["住","西雅图","机场","附近","酒店","有","哪些"]
]
```

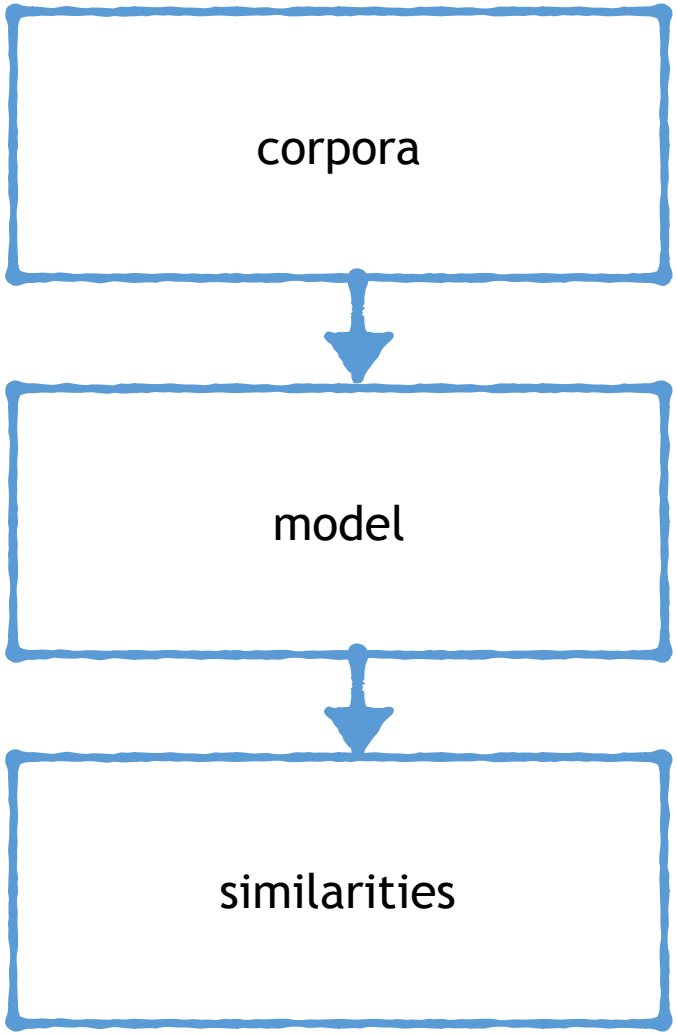
代码

```
from gensim import  corpora,similarities,models,matutils
import jieba
sentences=[]
stop_words = [', ', '。', '? ', '.', ',', '.', '的', ' ', '\n']
with open('./questions.txt') as f:
    for line in f.readlines():
        sentences.append([word for word in jieba.cut(line) if word not in stop_words])
dictionary=corpora.Dictionary(sentences)
print(dictionary.token2id["西雅图"])
```

注意设置停用词

GenSim的用法

```
from gensim import  corpora,similarities,models
```



Dictionary

Bow

TF-IDF

LDA　　HDP　　LSI

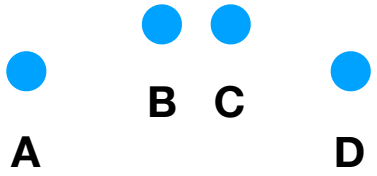
similarities.Similarity

similarities.MatrixSimilarity

similarities.SparseMatrixSimilarity

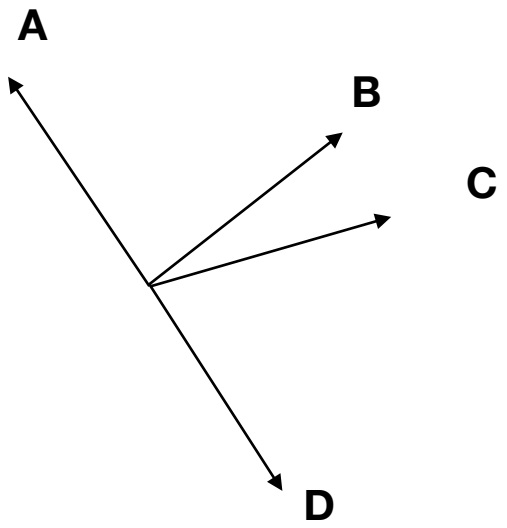
自然语言的相似计算

如何计算相似



计算点的相近用距离

$$\sqrt{(x2-x1)^2+(y2-y1)^2}$$



计算有向线段的相近用余弦夹角

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2+y_1^2} \times \sqrt{x_2^2+y_2^2}}$$

GenSim的使用

```
gensim.matutils.cossim(bow1,bow2)
```

这里传参bow 可以为简化表示的bow

待优化问题

1， 词没有权重

2， 词没有相关性， [1 0 0 0 0 0] [0 0 1 0 0 0 0]

3， 词向量太稀疏

继续优化

TFIDF算法

词频:（Term Frequency，缩写为TF）

逆文档频率（Inverse Document Frequency，缩写为IDF）

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

出现次数越多权重越大

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数}+1}\right)$$

越专有权重越大

$$\text{TF}-\text{IDF} = \text{词频(TF)} \times \text{逆文档频率（IDF）}$$

BOW结合TFIDF

```
models.TfidfModel(sentences_bow)
```

sentences_bow　为语料的bow

训练后的模型，可以向模型传递一句话的bow，返回这句话中每个词的tfidf权重

实现找相似模型

simlarity=similarities.MatrixSimilarity(bow,num_best=10,num_features=len(dictionary))

第一个参数bow 是一个二维的bow语料 ，[(5,3),(8,2),(10,3)]

num_best 表示返回多少个找到的相似的结果

num_features 是词典长度， 内部会根据这个参数生成稀疏矩阵

找相似：

simlarity[bow]， 这里的bow是一维的句子的bow向量

---

class `gensim.similarities.docsim.Similarity`(*output_prefix*, *corpus*, *num_features*, *num_best=None*, *chunksize=256*, *shardsize=32768*, *norm='l2'*)

Compute cosine similarity of a dynamic query against a static corpus of documents ("the index").

计算两篇文章的相似

所有的词编码

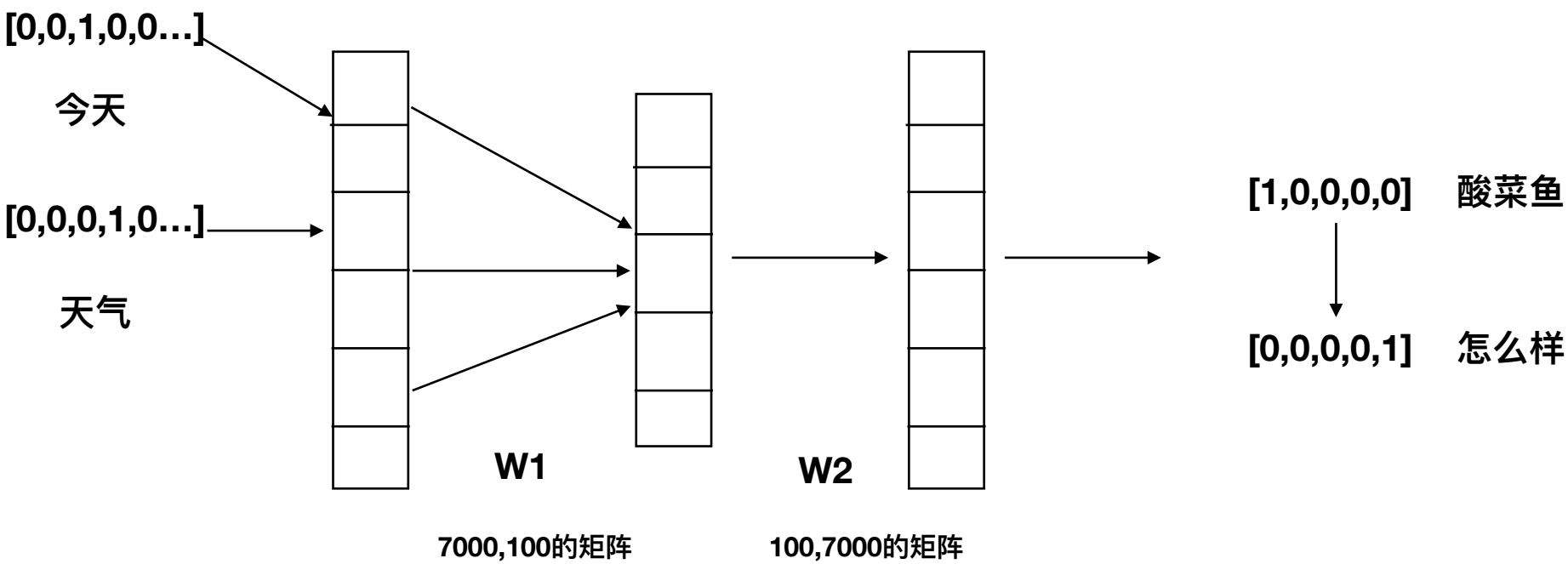
	第一篇文章	第二篇文章
中国	0.01	0
北京	0	0.02
蜜蜂	0.05	0
蜂蜜	0	0.04
养殖	0.04	0
产量	0	0.02

Word2Vec

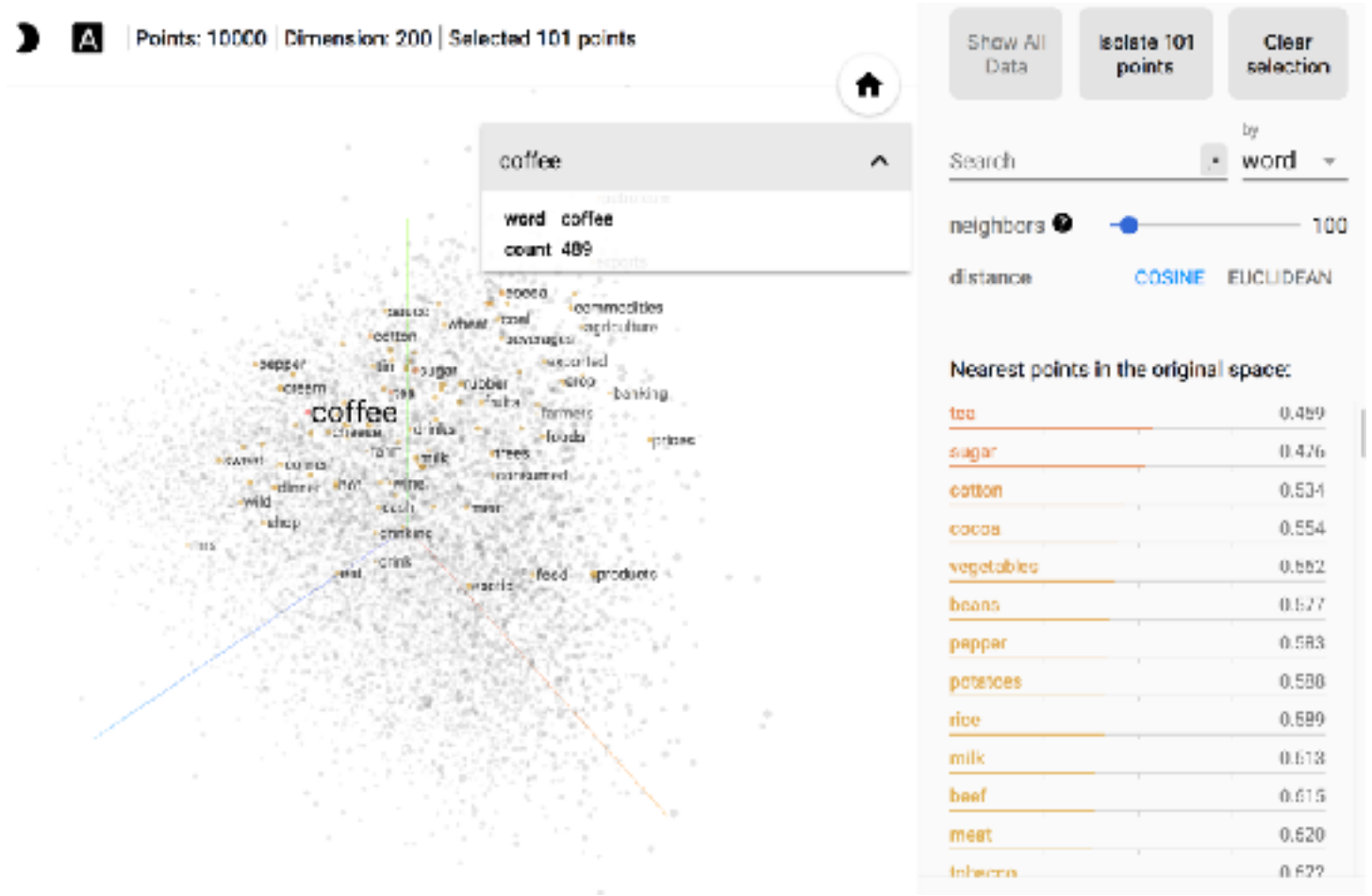
(N,7000) * (7000,100) = (N,100) (100,7000)

神经网络的出现

将一千维的向量，变为一百维



Word2Vec可视化演示



http://projector.tensorflow.org/

GenSim训练Word2Vec

gensim.model.Word2vec(sentences,100)

第一个参数是分词好的原始语料 。

第二个参数是特征维度

思考一下

OneHot -> BOW —> TFIDF —>Word2vec

word2vec 还能干什么？

购买买记录

《西游记》[1 0 0 0 0]

《红楼梦》 [0 1 0 0 0]

《三国演义》

....

HelloCode

开启智慧之旅