深度之眼
deepshare.net

词向量及Word2vec简介

导师: GAUSS

深度之眼
deepshare.net

目录

深度之眼
deepshare.net

词向量

Word Vectors

词向量

Word Vectors

one-hot 词向量

$$w_{\text{“男人”}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad w_{\text{“女人”}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad w_{\text{“小狗”}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad w_{\text{“小猫”}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$(w_{\text{“男人”}})^T w_{\text{“女人”}} = (w_{\text{“男人”}})^T w_{\text{“小狗”}} = 0$$

深度之眼
deepshare.net

文本表示方法

文本表示哪些方法？

基于one-hot、tf-idf、textrank等的bag-of-words；

主题模型：LSA（SVD）、pLSA、LDA；

基于词向量的固定表征：word2vec、fastText、glove

基于词向量的动态表征：elmo、GPT、bert

文本表示方法

One-hot 表示 ： 维度灾难、语义鸿沟；

分布式表示 (distributed representation) ：

- 矩阵分解（LSA）：利用全局语料特征，但SVD求解计算复杂度大；
- 基于NNLM/RNNLM的词向量：词向量为副产物，存在效率不高等问题；
- word2vec、fastText：优化效率高，但是基于局部语料；
- glove：基于全局预料，结合了LSA和word2vec的优点；
- elmo、GPT、bert：动态特征；

深度之眼
deepshare.net

Word2vec词向量

Word2vec based Word Vectors

Word2vec简介

Word2vec Introduction

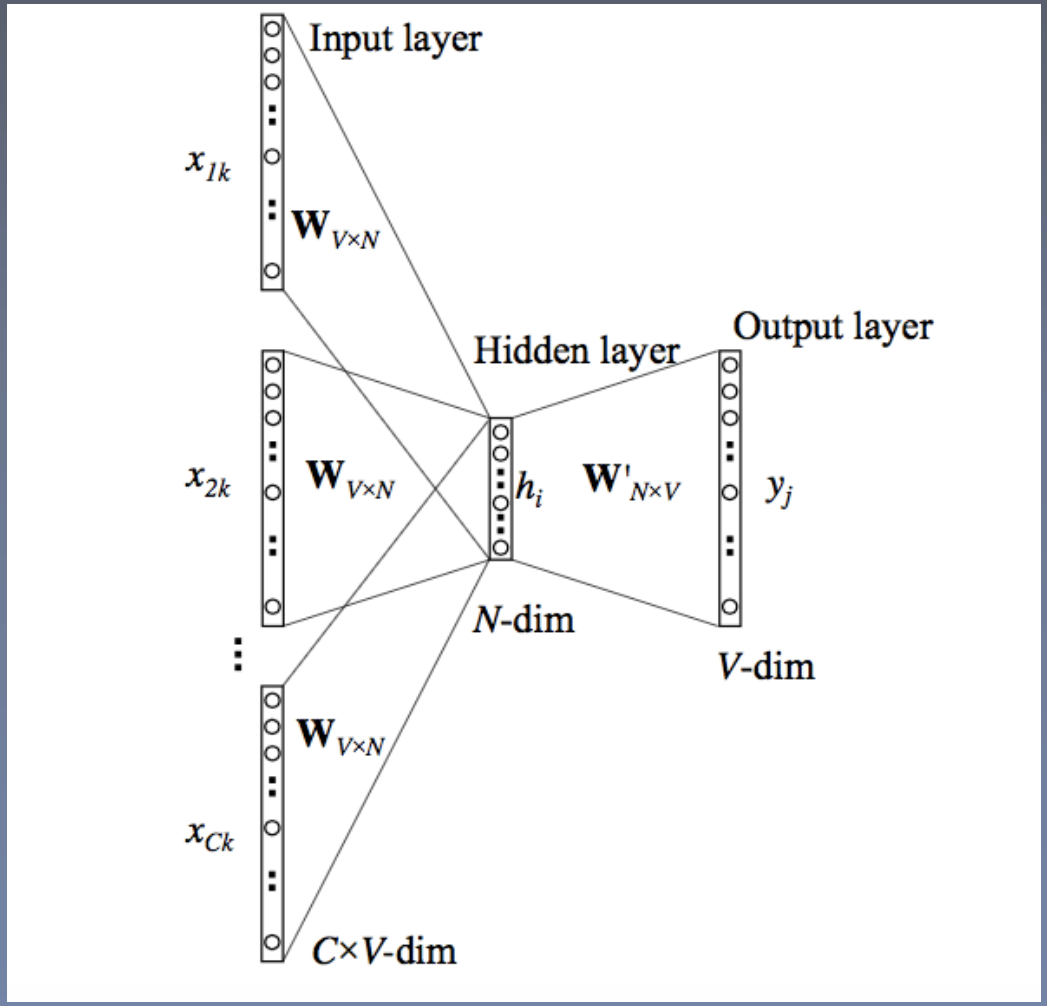
通俗的来讲，word2vec就是把 x 看做一个句子里的一个词语，y 是这个词语的上下文词语，那么这里的 f，便是 NLP 中经常出现的『语言模型』（language model），这个模型的目的，就是判断 (x,y) 这个样本，是否符合自然语言的法则，更通俗点说就是：词语x和词语y放在一起，是不是人话。

- 两个算法：CBOW 和 skip-gram
- 两种训练方法：负采样和层级softmax

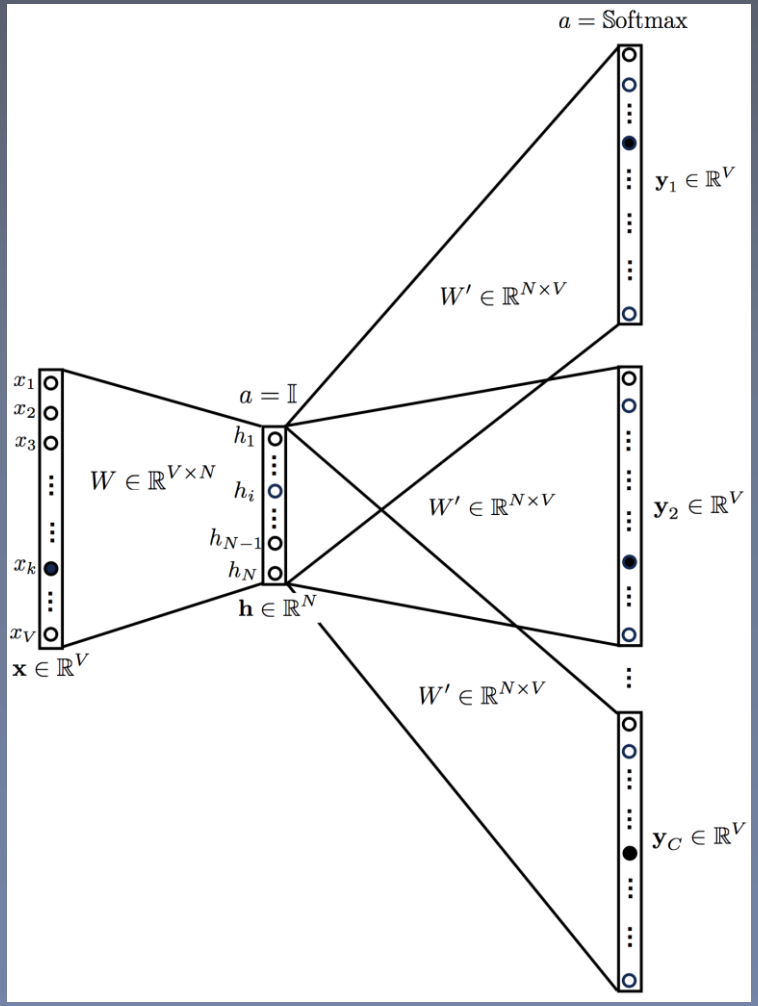
Word2vec简介

Word2vec Introduction

CBOW:



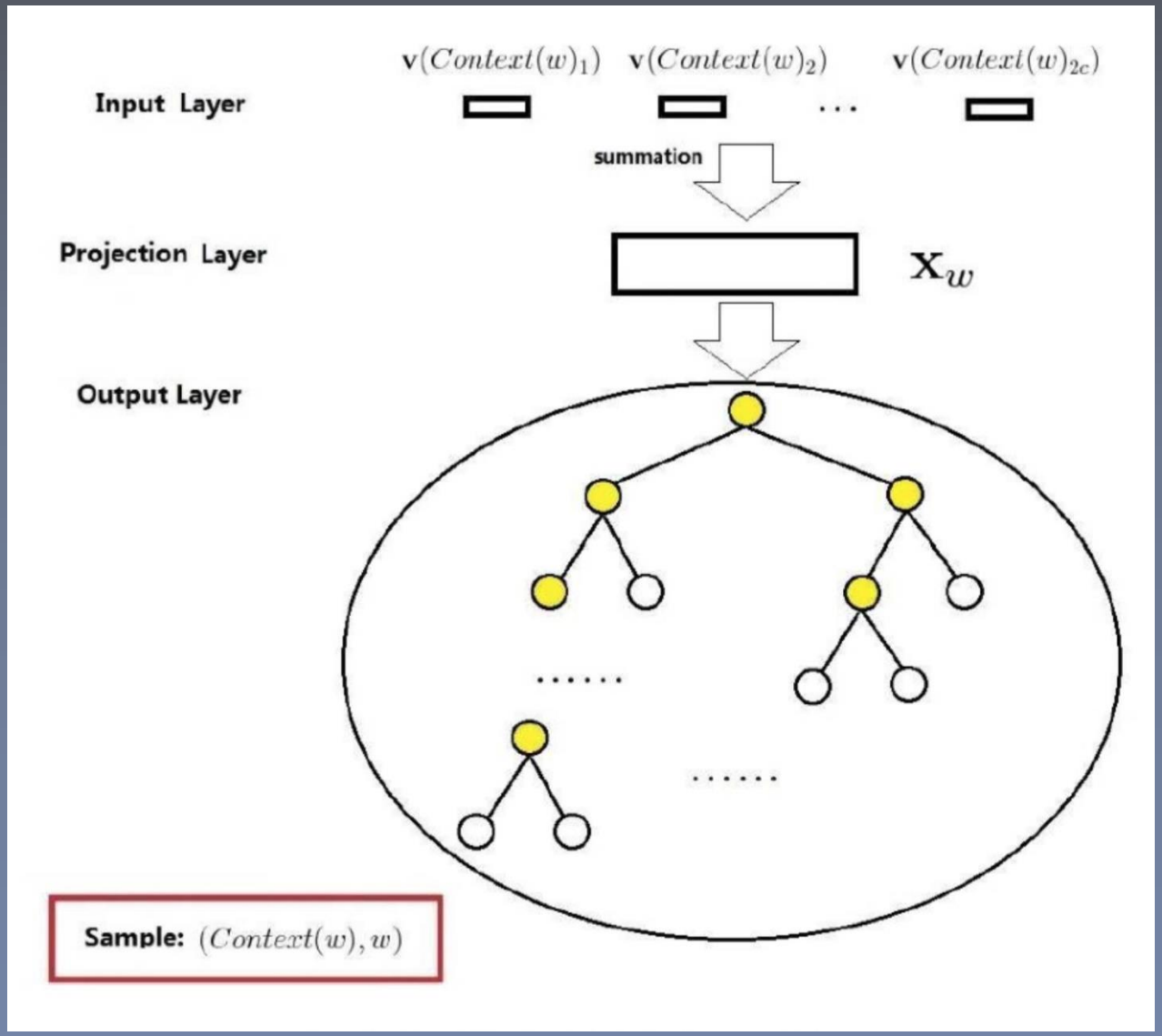
skip-gram:



Word2vec简介

Word2vec Introduction

层级softmax(Hierarchical Softmax)的网络结构图



Word2vec简介

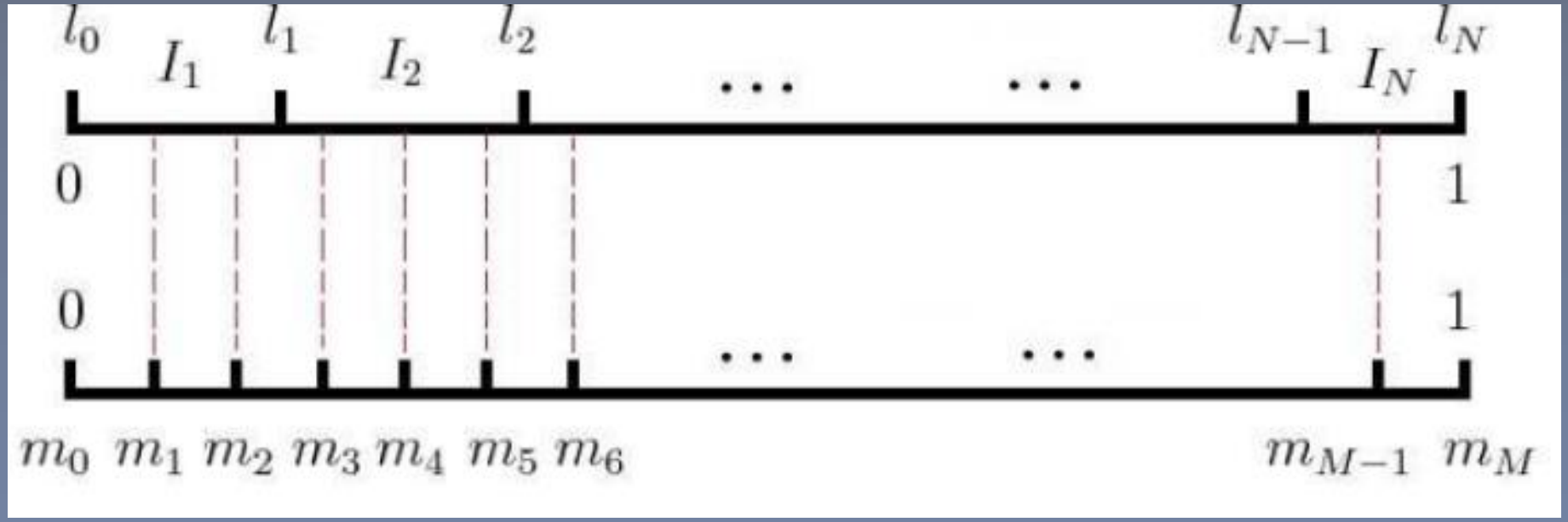
Word2vec Introduction

负采样的网络结构图

词汇表大小为N

将其分为M等份

M>>N



Word2vec简介

Word2vec Introduction

两者哪个更好呢？

- Skip-gram可以很好的处理少量数据，并且可以很好的表示稀疏单词。
- CBOW速度更快，对于更频繁的单词具有更好的表示。

训练方法哪个更好呢？

- 层级softmax 对低频词效果更好；
- 对应的负采样对高频词效果更好，向量维度较低时效果更好。

Word2vec使用

Word2vec Usage


```
!pip install --upgrade gensim

from gensim.test.utils import common_texts, get_tmpfile
from gensim.models import Word2Vec

path = get_tmpfile("word2vec.model")
model = Word2Vec(common_texts, size=100, window=5, min_c
ount=1, workers=4)

model.save("word2vec.model")
model = Word2Vec.load("word2vec.model")
vector = model.wv['computer']
```

Word2vec的参数详解

Details of the parameters of Word2vec

```
Word2Vec(sentences=None,  ---------->输入句子，可以是list格式
         size=100,  ---------->词向量的维度
         alpha=0.025,  ---------->初始学习率
         window=5,  ---------->句子中当前词与预测词之间的最大距离
         min_count=5,  ---------->忽略总频率低于此的所有单词
         sample=0.001,---------->对高频词进行下采样
         seed=1,  ---------->随机种子
         workers=3,  ---------->使用的线程数
         min_alpha=0.0001,  ---------->最小学习率
         sg=0,  ---------->训练算法，1为skip-gram,0为CBOW
         hs=0,  ---------->训练方法，1为层级softmax,0为负采样
         negative=5,  ---------->负采样的样本数
         ns_exponent=0.75,  ---------->用来形成负抽样分布的指数
         cbow_mean=1,---------->如果为0，则使用上下文单词向量的和。如果为1，使用平均值
         iter=5,  ---------->语料库上的迭代次数,
         sorted_vocab=1,---------->如果是1，在分配单词索引之前，按降序频率对词汇表进行排序
         compute_loss=False---------->是否计算并存储loss值
         )
```

实操！！！

深度之眼
deepshare.net

总结

本节小结

Summary

词向量及Word2vec简介	词向量	
	文本表示方法	
	Word2vec词向量	

结语

我 说

看过千万代码，不如实践一把！

深度之眼
deepshare.net

联系我们：

电话：18001992849

邮箱：service@deepshare.net

Q Q： 2677693114

公众号

客服微信