



使用RNN建模序列 与Demo

PRESENTED BY Wei Lai
weilai5@jd.com

OUTLINE

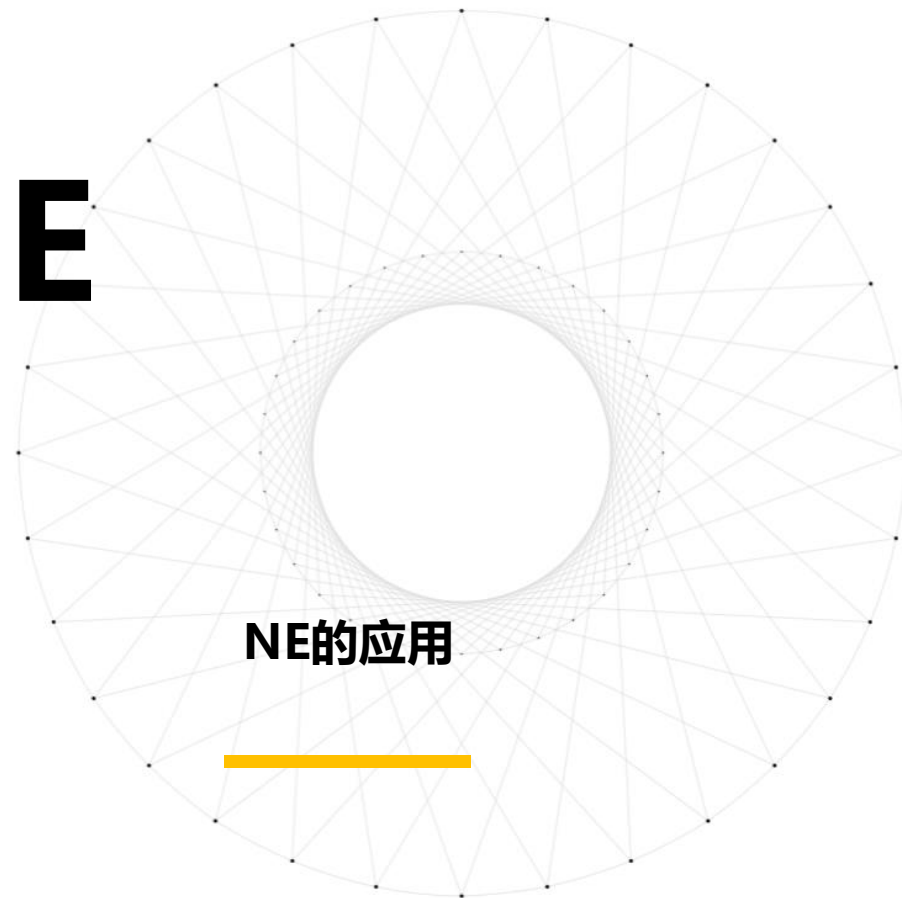
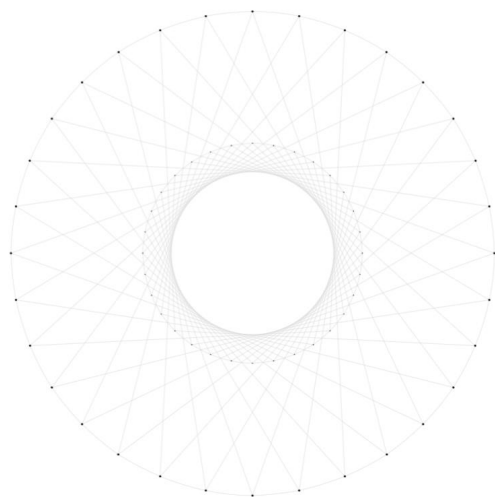
序列建模

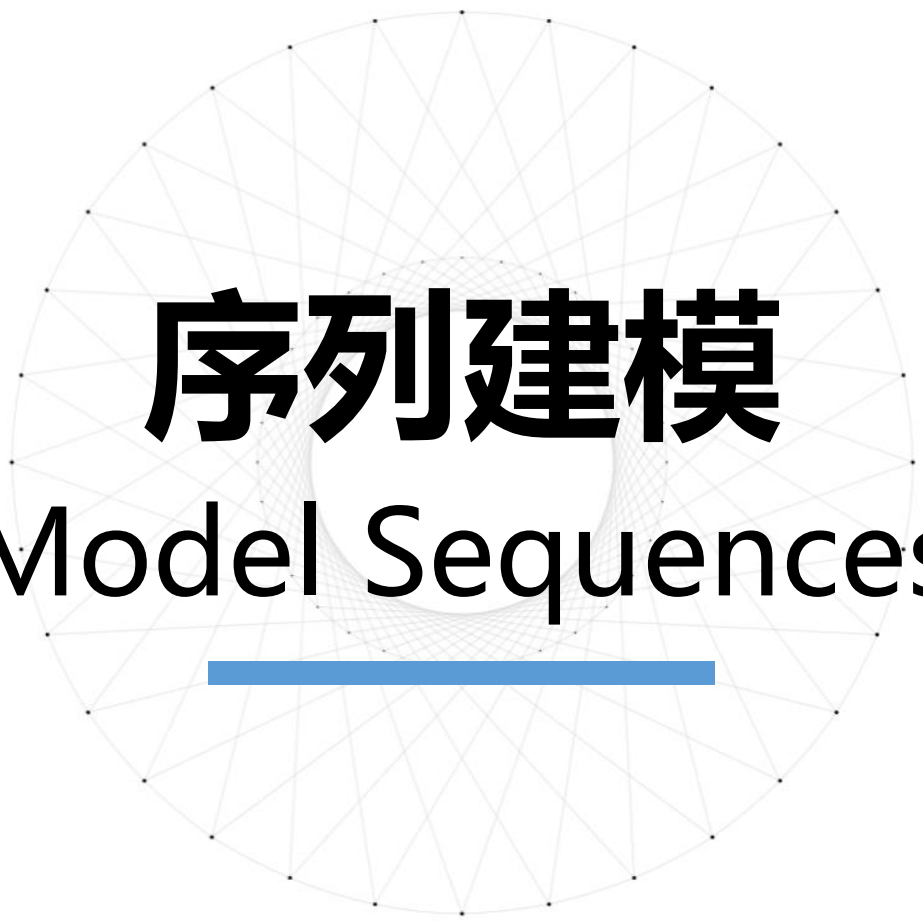


网络数据的表示学习



NE的应用





序列建模

Model Sequences

序列数据

词序列 ：自然语言

行为序列：用户在JD上先后浏览过的页面、商品和商铺等

人工构造的序列：在一个Graph上随机游走，得到的图节点序列

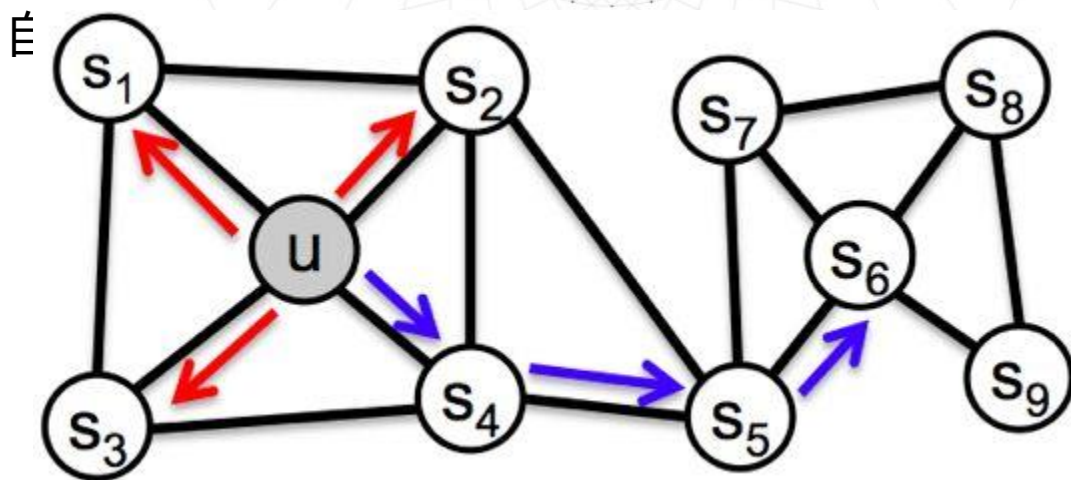
[我, 爱, 机器学习]

[这, 菜, 真, 香, 我, 还, 想, 吃]

[Deep, is, powerful, in, feature, learning]

User1: [page1, page2, page3]

User2: [page10, page2, page3]



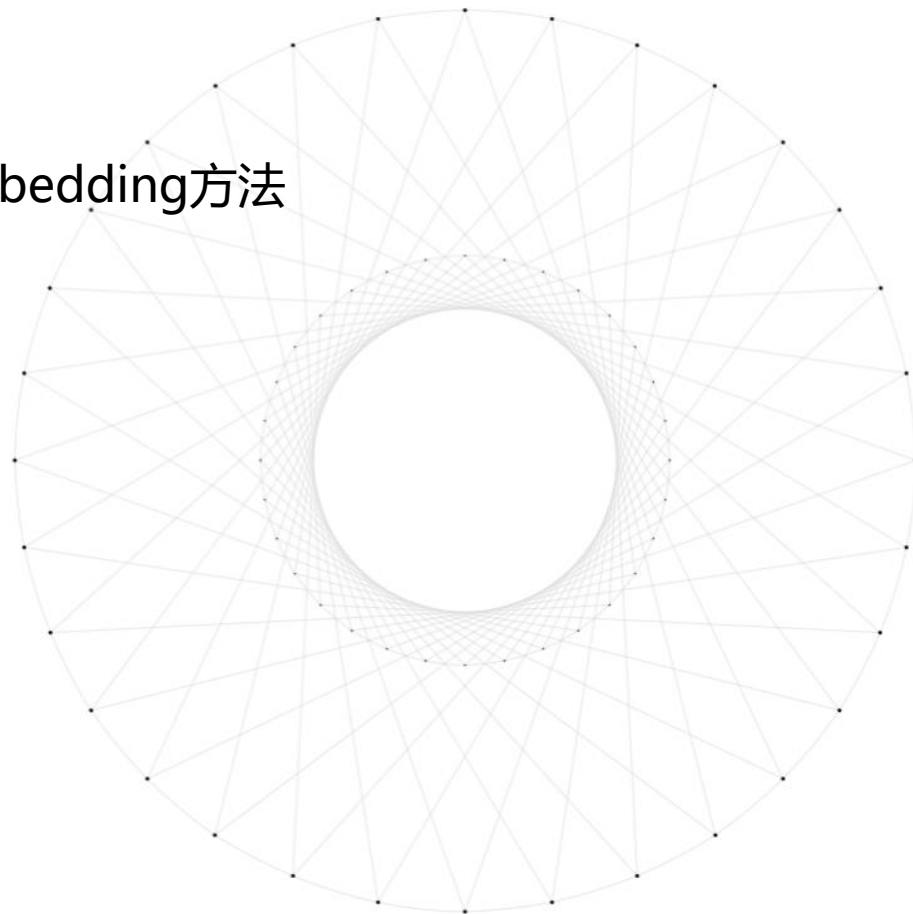
建模序列

学习序列中节点的特征表示:

词向量, 部分Network Embedding方法

学习整个序列的特征表示:

文本分类, 机器翻译



用RNN建模序列数据

Sequence: $(x_0, x_1, x_2, x_3 \dots x_n)$

A RNN Cell $f(h, x)$:

$$h_{t+1} = f(h_t, x_{t+1})$$

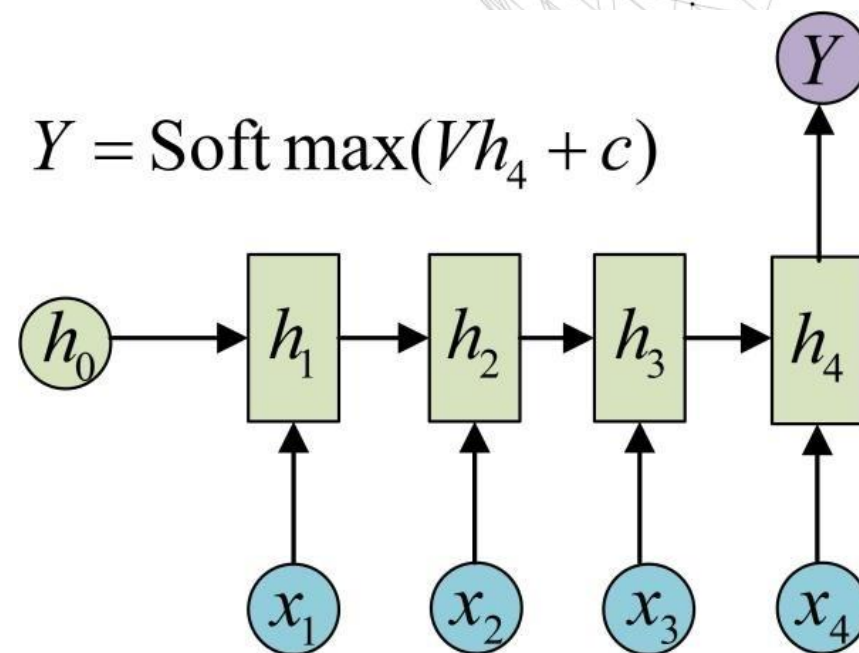
$$f(h_t, x_{t+1}) = W_h h_t + W_x x_{t+1}$$

Initialize h_0 randomly.

for t in range(n):

$$h_{t+1} = f(h_t, x_t)$$

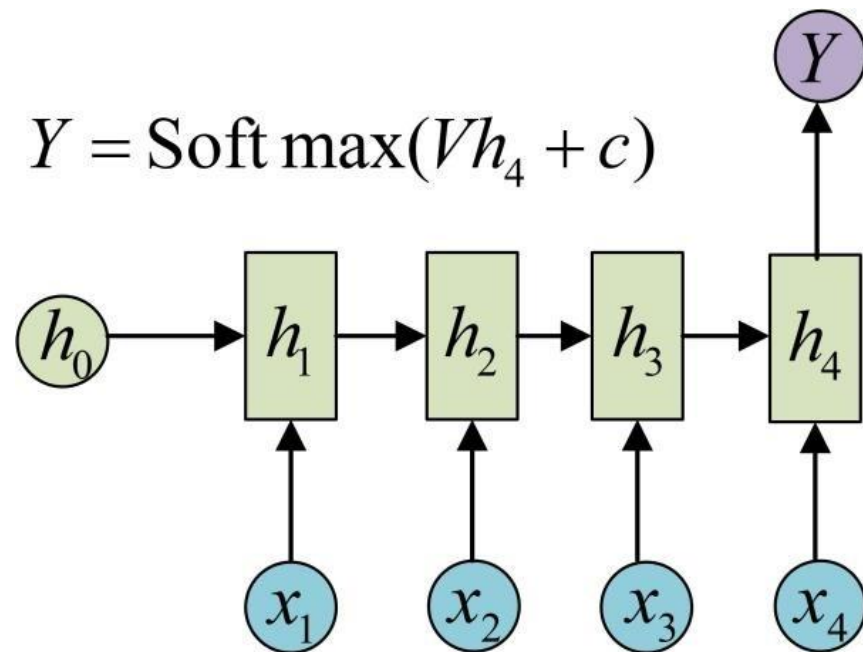
最后一个隐含状态 (hidden state) 包含了整个序列的信息



梯度异常与长期依赖

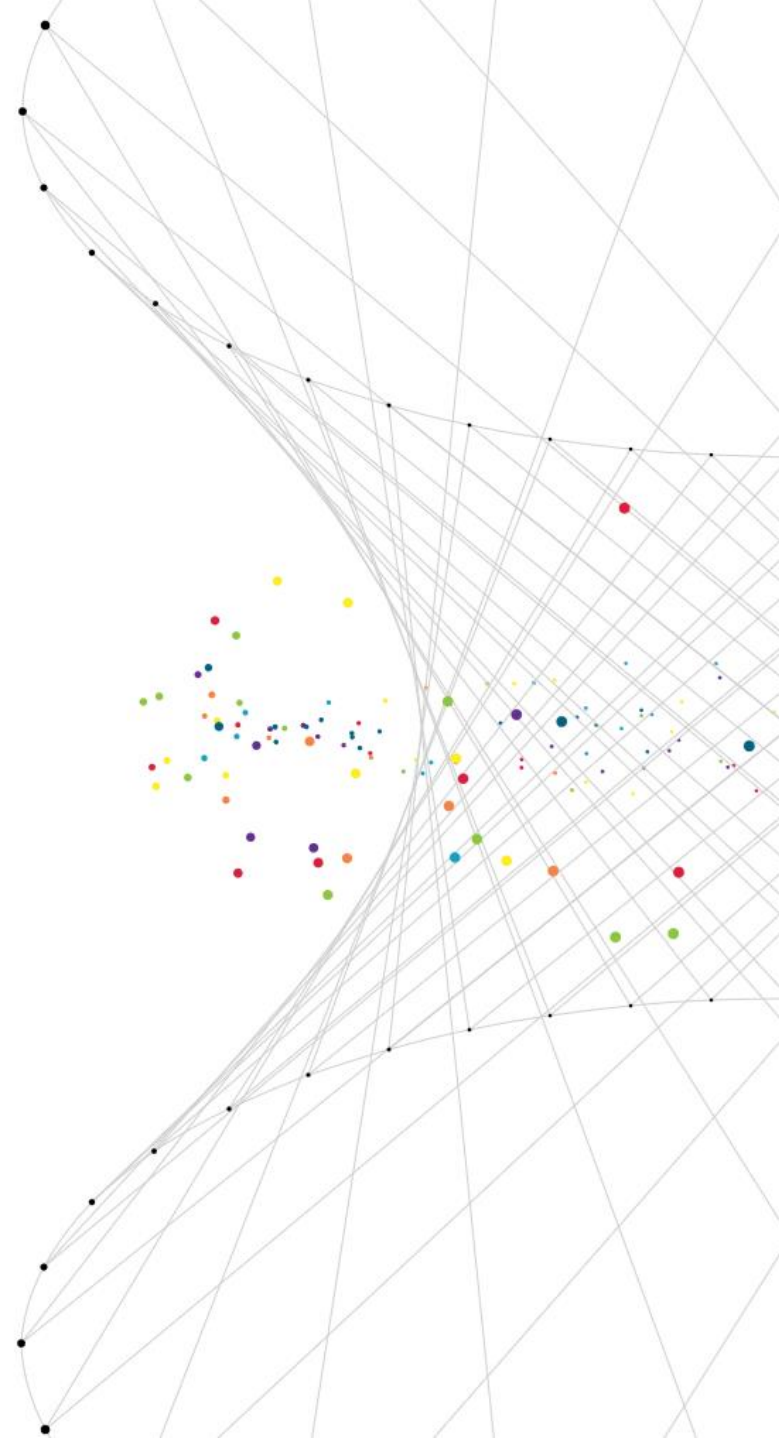
梯度爆炸与消失 导致模型难以训练

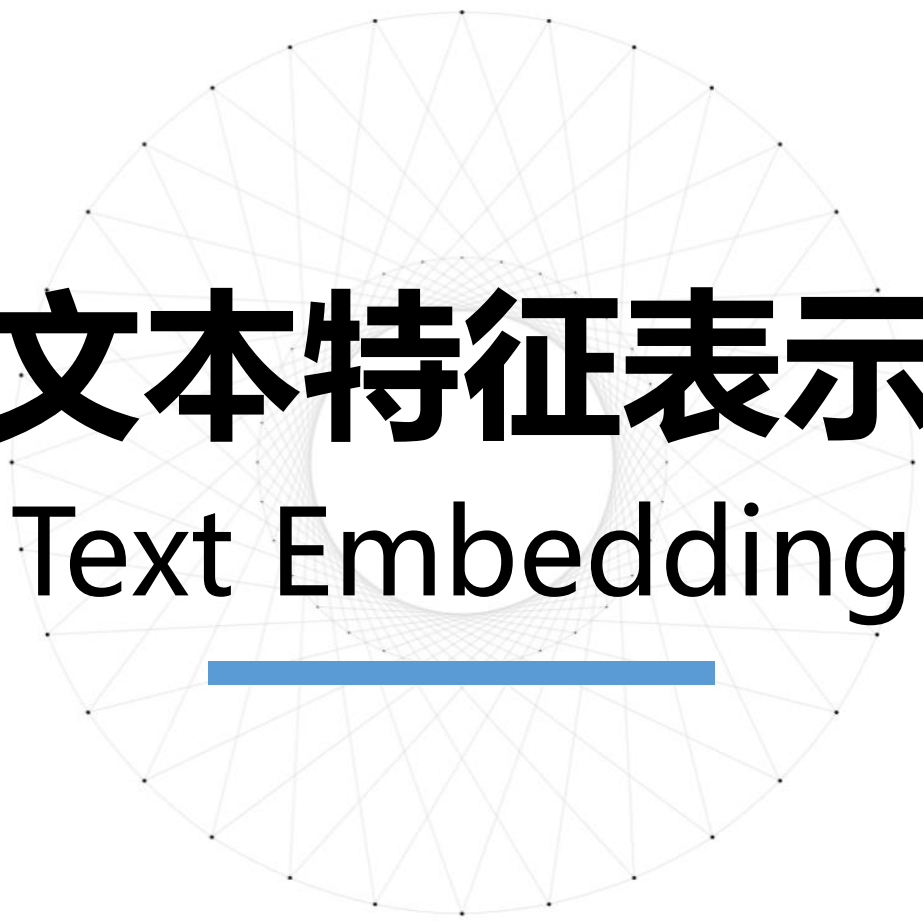
传统的RNN Cell无法很好的捕捉长期依赖



Embedding特征

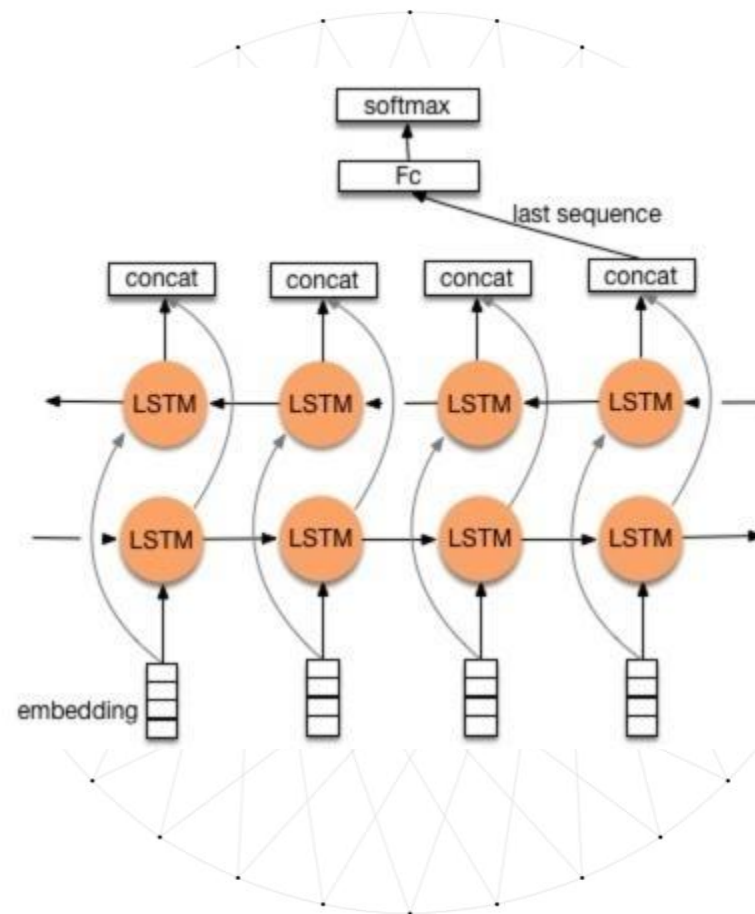
用低维、连续的嵌入空间表示词、元素等任意实体。

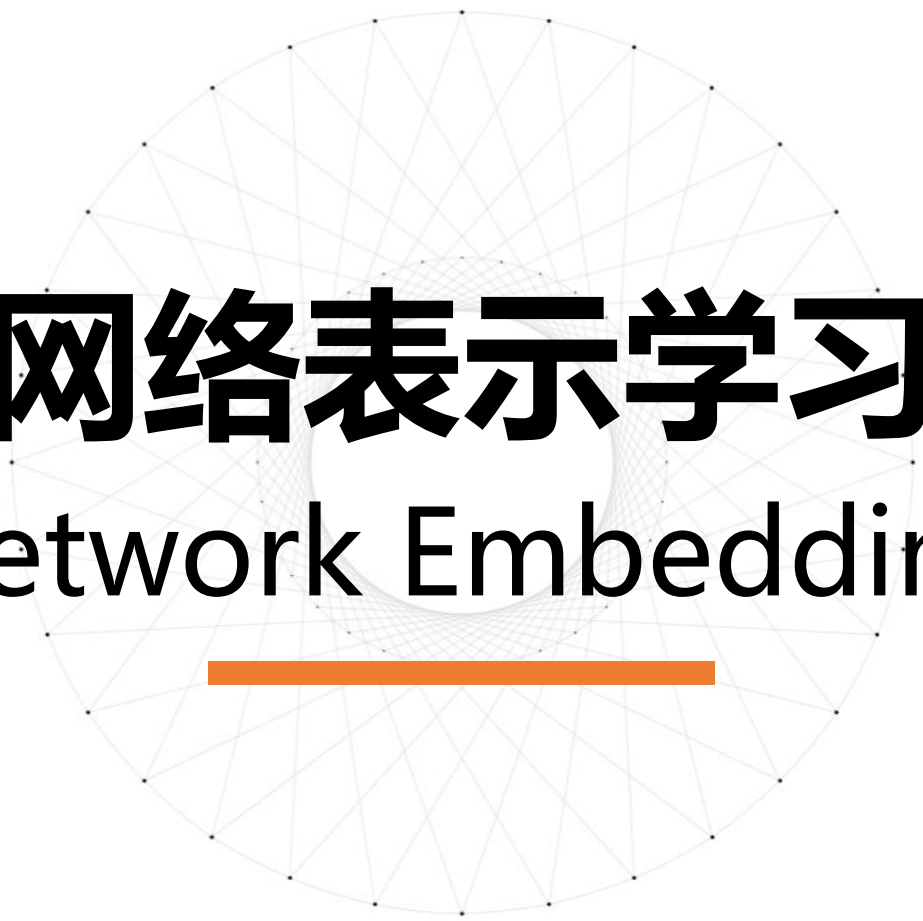




文本特征表示

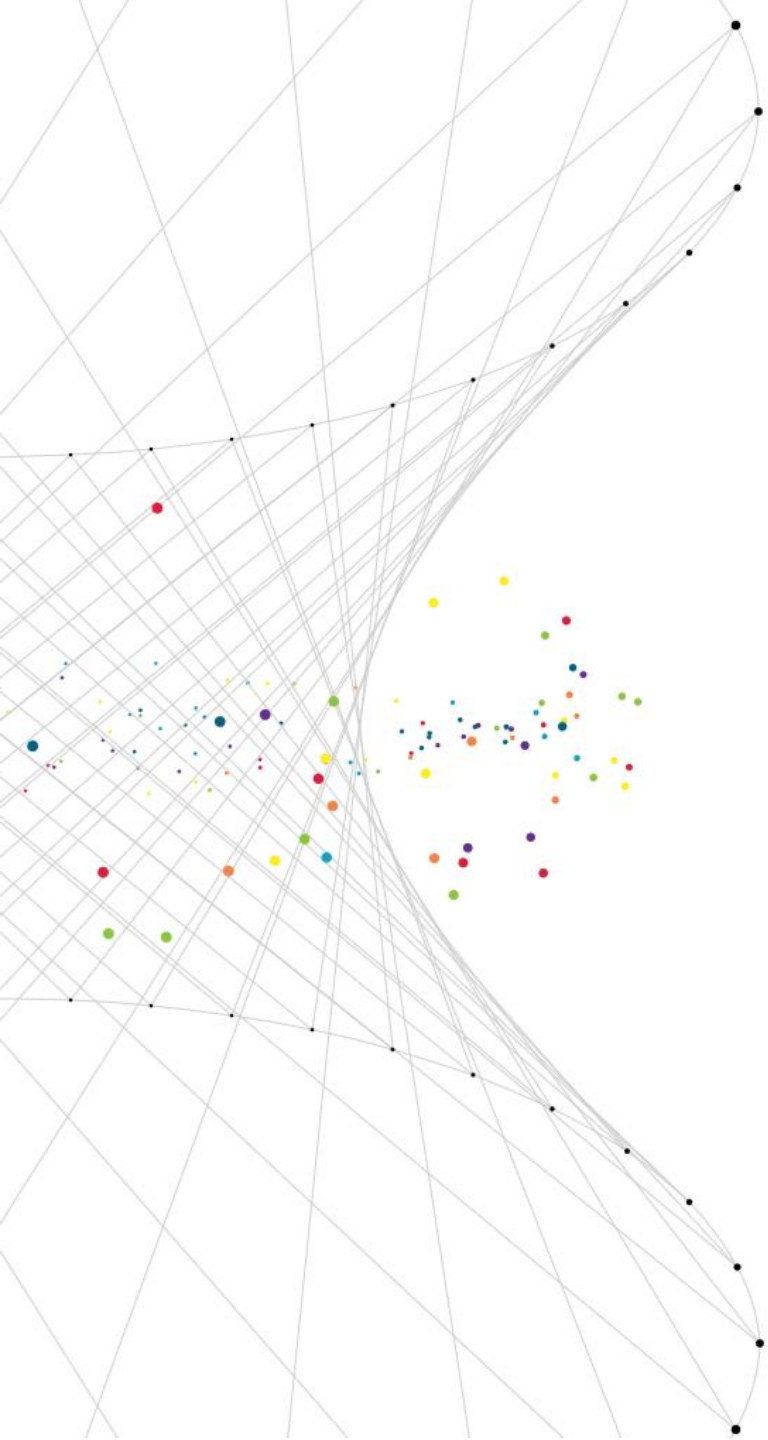
Text Embedding





网络表示学习

Network Embedding



原生的网络数据在现实生活中非常常见：

- 由好友关、关注被关注产生的社交网络
- 由论文引用、学者合作关系产生的引用、学者合作的网络
- 知识图谱中的 实体-关系 也是一个网络

同时，我们还可以强行构建网络：

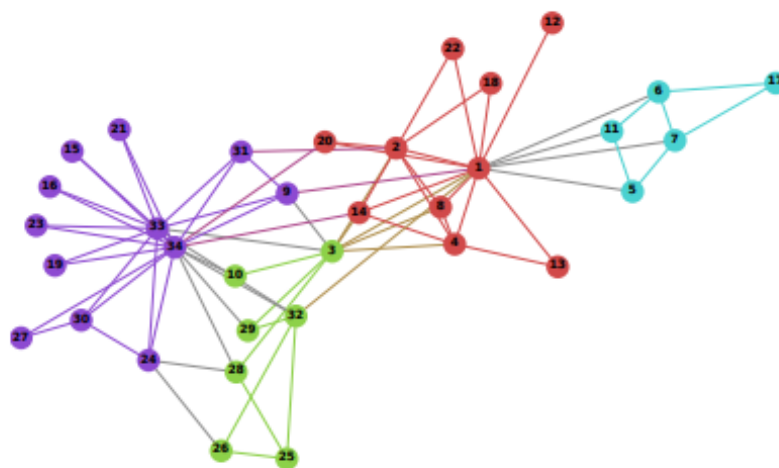
- 多标签数据集中，标签与同一个样本共现的关系，可以构建标签网络
- 电商网站的购物车-共同购买的关系网络
- 以word-context的局部共现关系，可以构建词共现网络；

简单说，有实体和实体间关系，我们就可以构建出一个网络

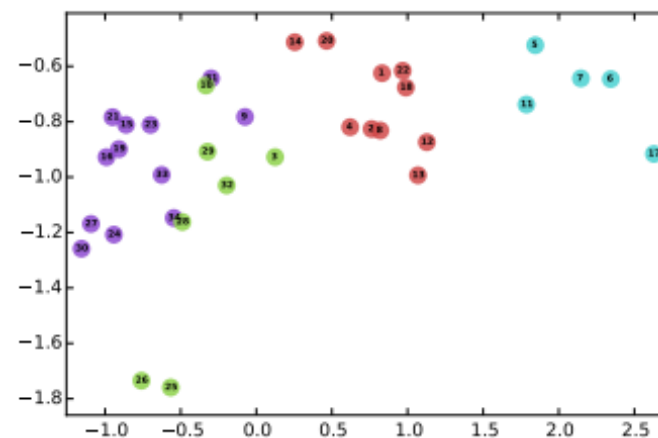
网络嵌入的目标

学习网络中节点或边的低维、连续的向量表示，使之保留网络中的结构信息。

学习到的特征表示，可以直接用于可视化、链接预测、节点分类等任务



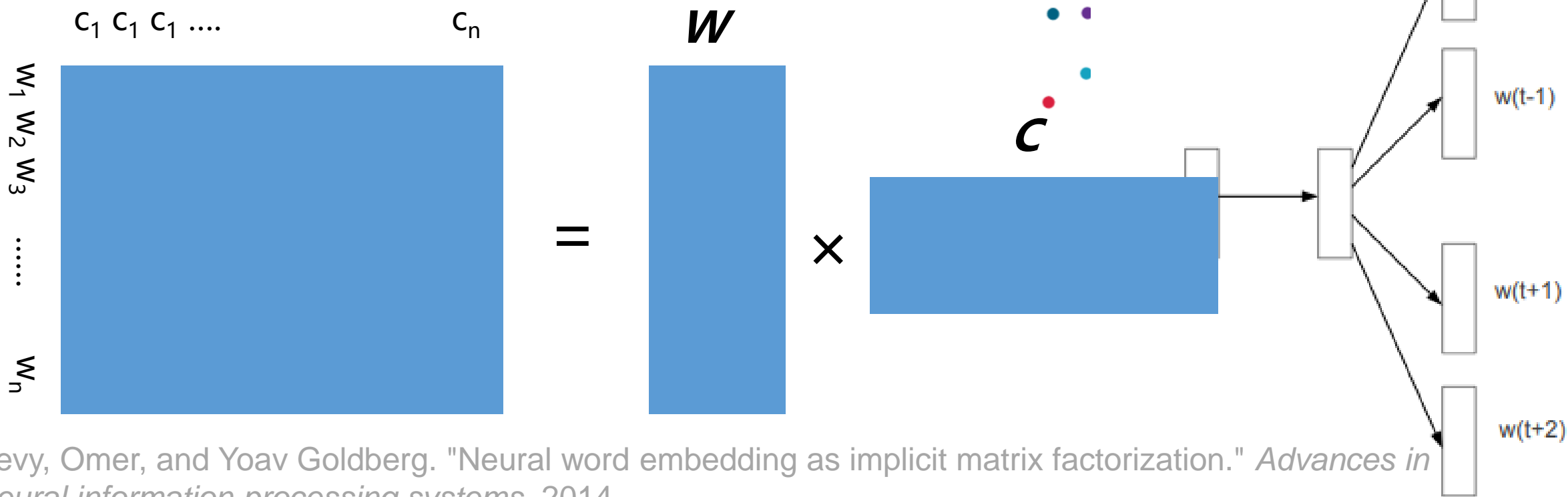
(a) Input: Karate Graph



(b) Output: Representation

Word2vec: Skip-gram negative sampling

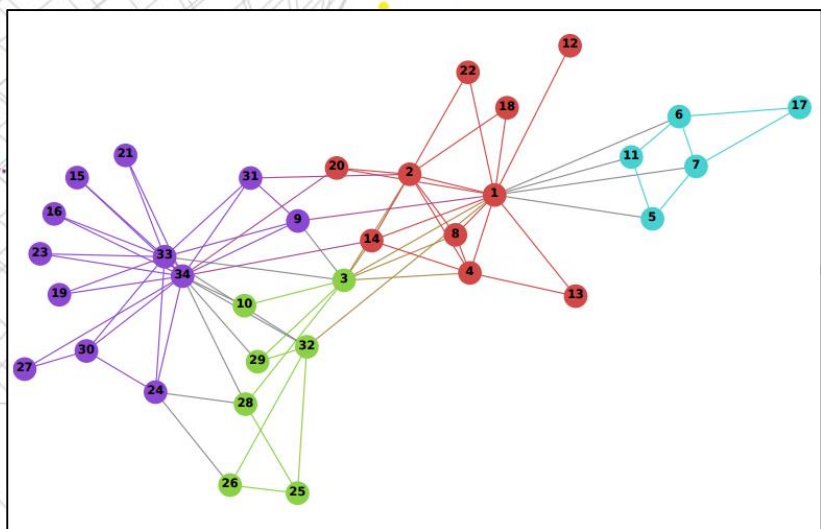
对word, context pair的似然最大化, 其物理意义就是预测一个词周围出现的词
可以看出, SGNS是根据文本中的局部共现进行建模, 除了概率模型,
我们还可以通过**矩阵分解**的形式获得类似效果的词向量



Levy, Omer, and Yoav Goldberg. "Neural word embedding as implicit matrix factorization." *Advances in neural information processing systems*. 2014.

经典模型——DeepWalk

既然SGNS可以看做是对一个**word-context共现的邻接矩阵**上做的操作
那么事实上，它也可以看做是对一个word的共现网络所做的操作



V1, V3, V6.....

V4, V1, V33...

V1, V3, V6.....

V4, V1, V33...

V1, V3, V6.....

V4, V1, V33...

V1, V3, V6.....

.....

WORD2VEC

V1: (0.22, 0.33, 0.44.....)

V2:(0.45,-0.1, 0.88....)

.....

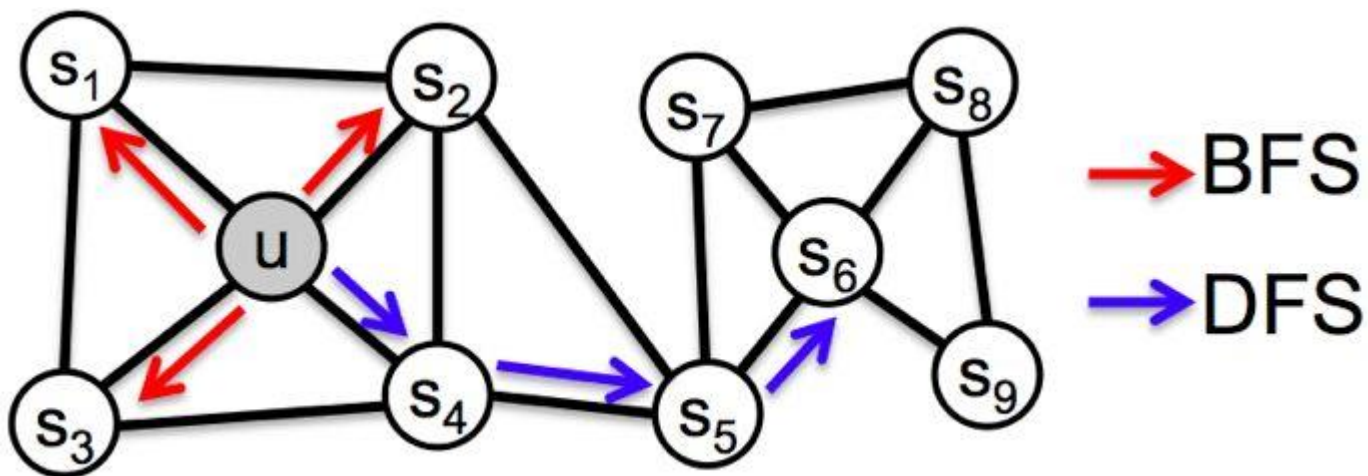
随机游走得到节点序列

将序列输入word2vec，得到节点特征向量

经典模型——Node2vec

Node2vec觉得DeepWalk的**随机游走**不好，无法考虑到更global的全局信息，所以结合了深度优先和广度优先游走来构建节点序列

得到序列后，仍然是使用word2vec中的词向量模型得到节点模型。

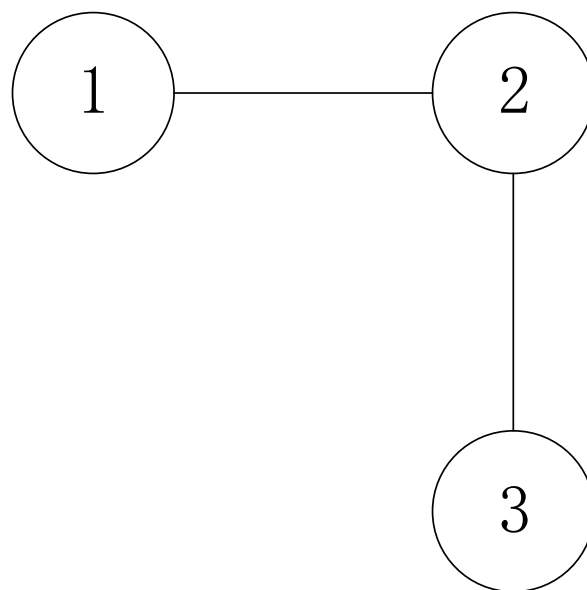


经典模型——LINE

网络中的关系分为**一阶亲密度**和**二阶亲密度**:

一阶亲密度: 两个节点之间有边直接相连: $(1, 2)$, $(2, 3)$

二阶亲密度: 两个节点共享同一个邻居: $(1, 3)$





经典模型——LINE

与SGNS类似，LINE也使用条件概率建模，核心是最大化经验分布（观测到的一阶or二阶阶亲密度的节点对）与所提模型的分布KL散度

一阶亲密度：两个节点直接相连

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}, \quad O_1 = d(\hat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot)),$$

二阶亲密度：两个节点通过某个节点相连

$$p_2(v_j | v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}, \quad O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot | v_i), p_2(\cdot | v_i)),$$