2020-2021学年秋季学期

自然语言处理

Natural Language Processing

授课教师： 胡玥

助　教： 于静

自然语言处理
Natural Language Processing

第 9 章　词向量

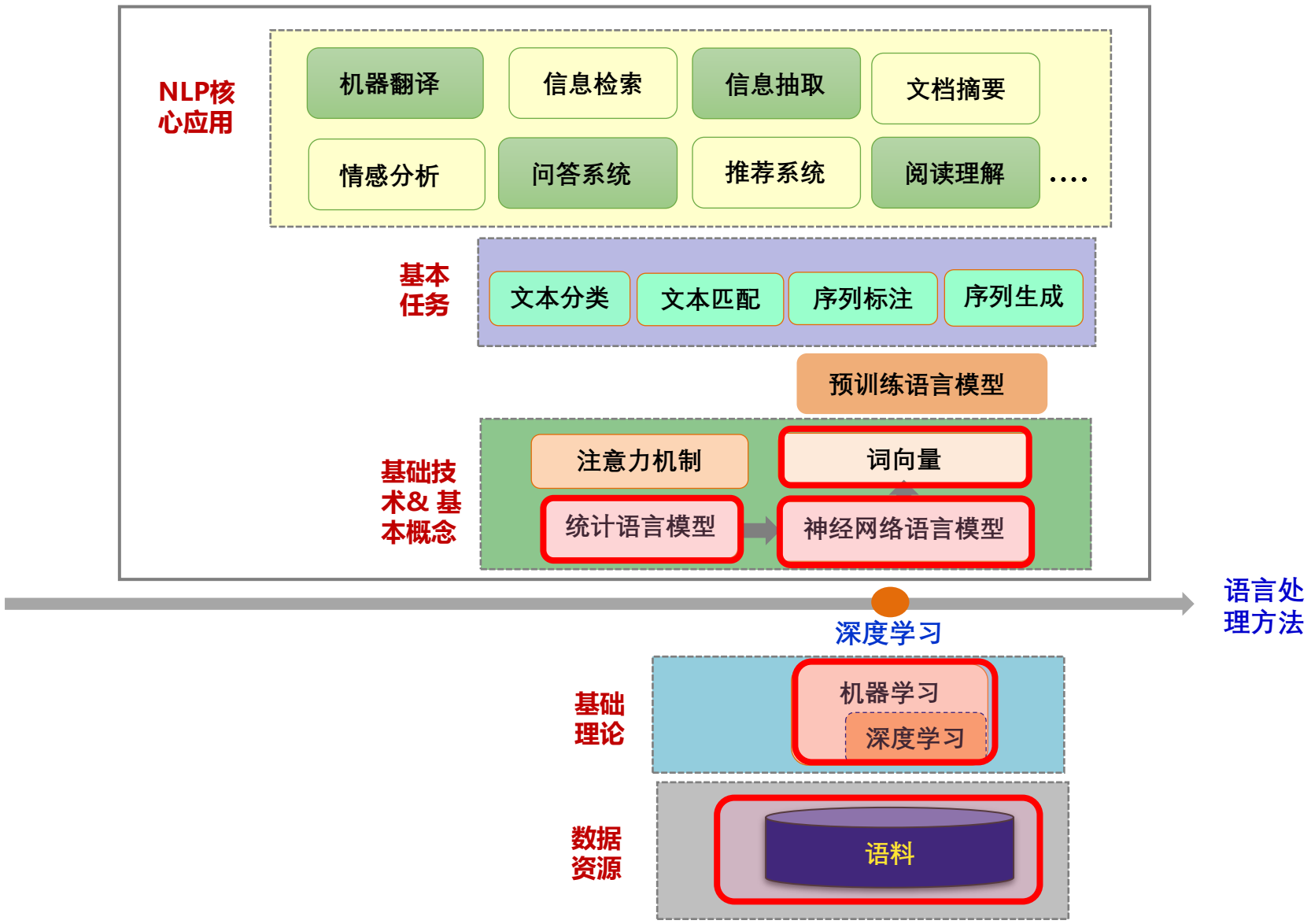
（浅层模型）

授课教师：　胡玥

授课时间：2020.9

课程编码 201M4005H　课程名称 自然语言处理　授课团队名单 胡玥 、于静

基于深度学习的自然语言处理课程内容



概 要

本章主要内容：

浅层词表示模型：介绍5种经典的词向量模型 NNLM模型(词向量)，RNNLM模型(词向量)， C&W模型，CBOW模型，Skip-gram模型 的模型结构及学习方法；

本章教学目的：

使学生了解词向量技术的发展过程及种各词向量技术的特点。

内 容 提 要

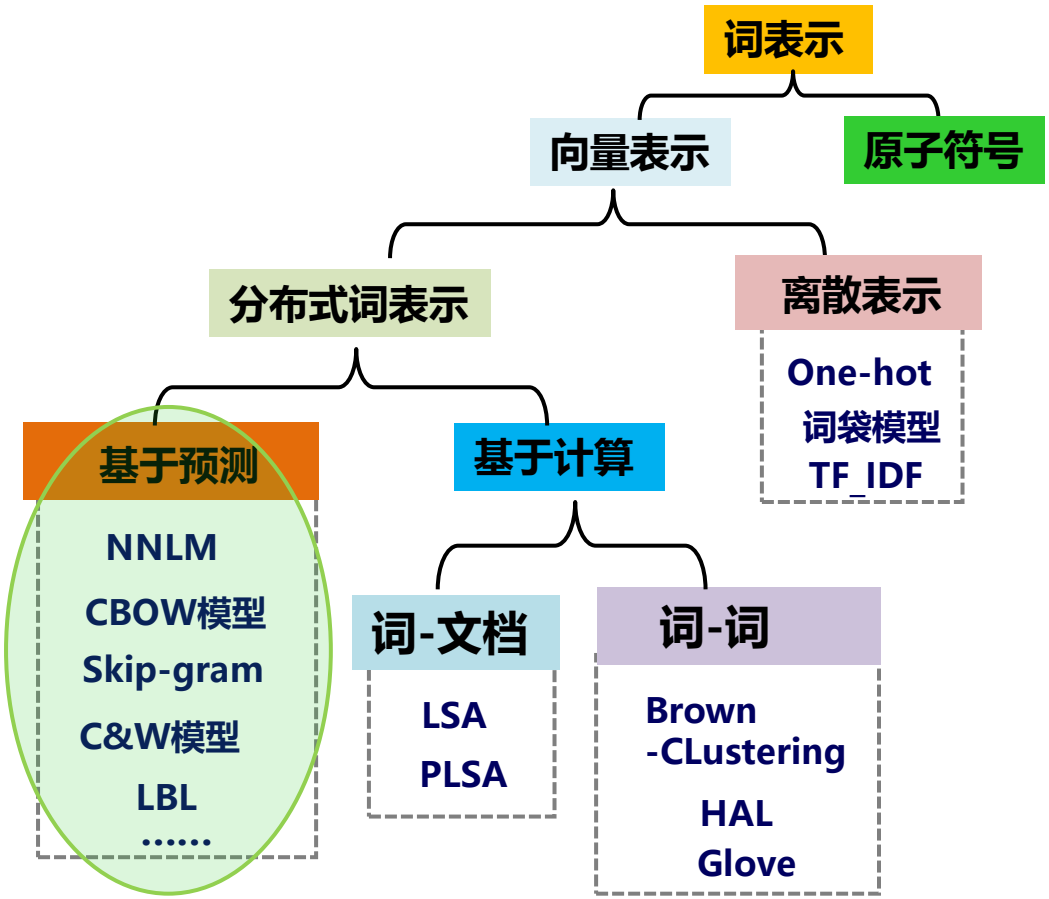
9.1　词表示概述

9.2　浅层词表示模型

9.1 词表示概述

词的表示

自然语言问题要用计算机处理时，第一步要找一种方法把这些符号数字化，成为计算机方便处理的形式化表示。



9.1 词表示概述

■ 符号表示

大部分基于规则和基于统计的自然语言处理任务把词看作**原子符号**

如，减肥　瘦身

■ 离散表示

1、One-hot 表示

NLP 中最直观，也是最常用的词表示方法

如：
- 减肥 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
- 瘦身 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

优势：稀疏方式存储非常的简洁

不足：词汇鸿沟，维数灾难

2、词袋模型

每个数表示该词在文档出现的次数（One-hot的加和）

3、TF_IDF

每个数代表该词在整个文档中的占比

■ 词的分布式表示

分布式假设：在相同上下文中出现的词倾向于具有相同的含义 [Harris ,1954]

如，Marco saw a hairy little **wampinuk** Crouching behind a tree .

wampinuk 的含义可由其上下文推断。

分布式语义学：根据词语在大型文本语料中的分布特性量化词语及词语语义相似性。

核心思想：用一个词附近的其他词来表示该词

9.1 词表示概述

经典分布表示模型

名称	上下文	上下文与目标词之间的建模（技术手段）
LSA/LSI HAL GloVe Jones & Mewhort	文档 词 词 n-gram	矩阵
Brown Clustering	词	聚类
Skip-gram CBOW Order LBL NNLM C&W	词 n-gram（加权） n-gram（线性组合） n-gram（线性组合） n-gram（非线性组合） n-gram（非线性组合）	神经网络

- **基于计算的分布式**

利用全部上下文或利用一定窗口内的上下文词捕获语法和语义信息

局限性：耗空间过大、稀疏等问题，需用降维方法（如，SVD分解的方法）构造低维稠密向量作为词的分布式表示 (25~1000维)。与深度学习模型框架差异大。

- **基于预测的分布式表示（神经网络词向量）**

不计算词之间的共现频度，直接用“基于词的上下文词来预测当前词”或“基于当前词预测上下文词”的方法构造构造低维稠密向量作为词的分布式表示 。

在深度学习自然语言处理技术中，词向量是神经网络技术中重要的组成部分

9.1 词表示概述

词向量(词嵌入)提出

2013年 - 词嵌入（Word Embeddings）— ★里程碑

词嵌入在2001年首次出现。而 Mikolov 等人在2013年作出的主要创新是通过删除隐藏层和近似目标来使这些单词嵌入的训练更有效。虽然这些变化本质上很简单，但它们与高效的 word2vec（word to vector，用来产生词向量的相关模型）组合在一起，使得大规模的词嵌入模型训练成为可能。

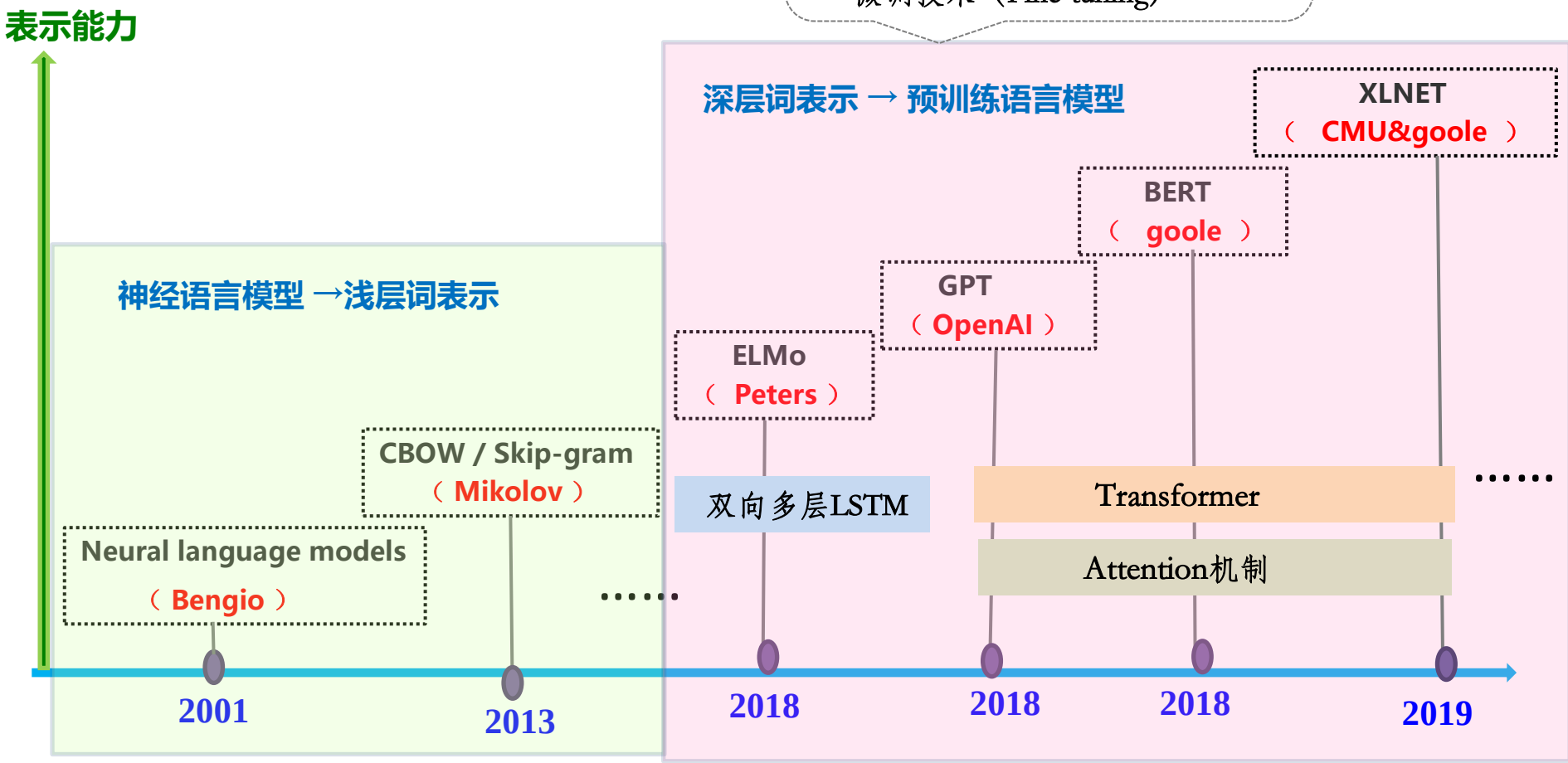
Paper：

Tomas Mikolov:Efficient estimation of word representations in vector space，2013

9.1 词表示概述

■ 预训练语言模型发展历程

相关概念
- 迁移学习（Transfer Learning）
- 多任务学习（Multi-task Learning）
- 微调技术（Fine-tuning）



神经语言模型 → 浅层词向量→ 深层词向量→预训练语言模型

内 容 提 要

9.2.1 经典词(向量)表示模型

1. NNLM模型词向量
2. RNNLM模型词向量
3. C&W 模型词向量
4. CBOW 模型词向量
5. Skip-gram模型词向量

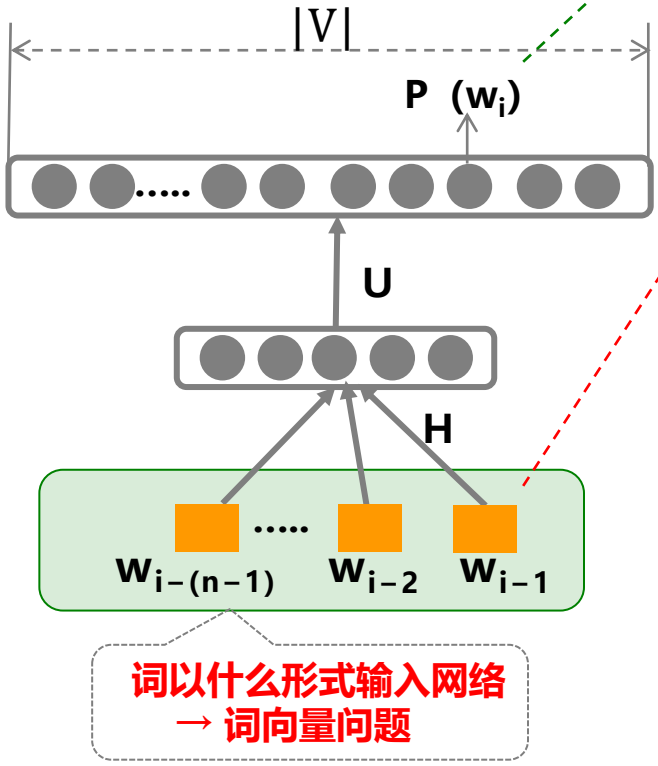
1. NNLM模型(词向量)

问题引入：NNLM模型（n-gram）回顾

语言模型参数

n-gram: $p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i|w_{i-(n-1)}\ \ldots\ w_{i-1})$

NNLM模型结构



softmax(y)

输出层：$p(w_i|w_{i-(n-1)}\ \ldots\ w_{i-1})=\frac{\exp(y(w_i))}{\sum_{k=1}^{|V|}\exp(y(v_k))}$

$y(w_i)=b^2+U(\tanh(XH+b^1))$

隐藏层：$h=\tanh(XH+b^1)$

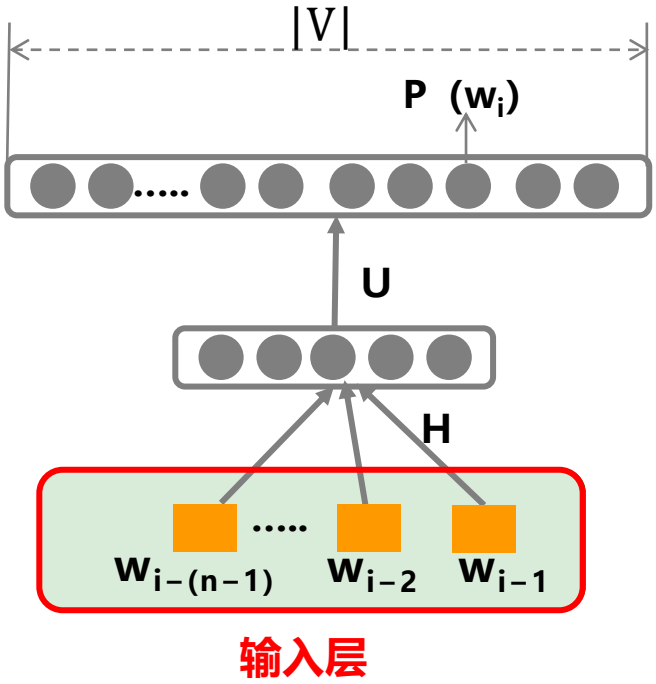
输入层：X: n-1个词 $w_{i-(n-1)},\ldots,w_{i-1}$

参数：$\theta=\{H,U,b^1,b^2\}$

神经网络参数

1. NNLM模型(词向量)

NNLM模型-输入表示



词的 one-hot 表示

张： 0000100.....00

三：0010000.....00

李：00000010.....00

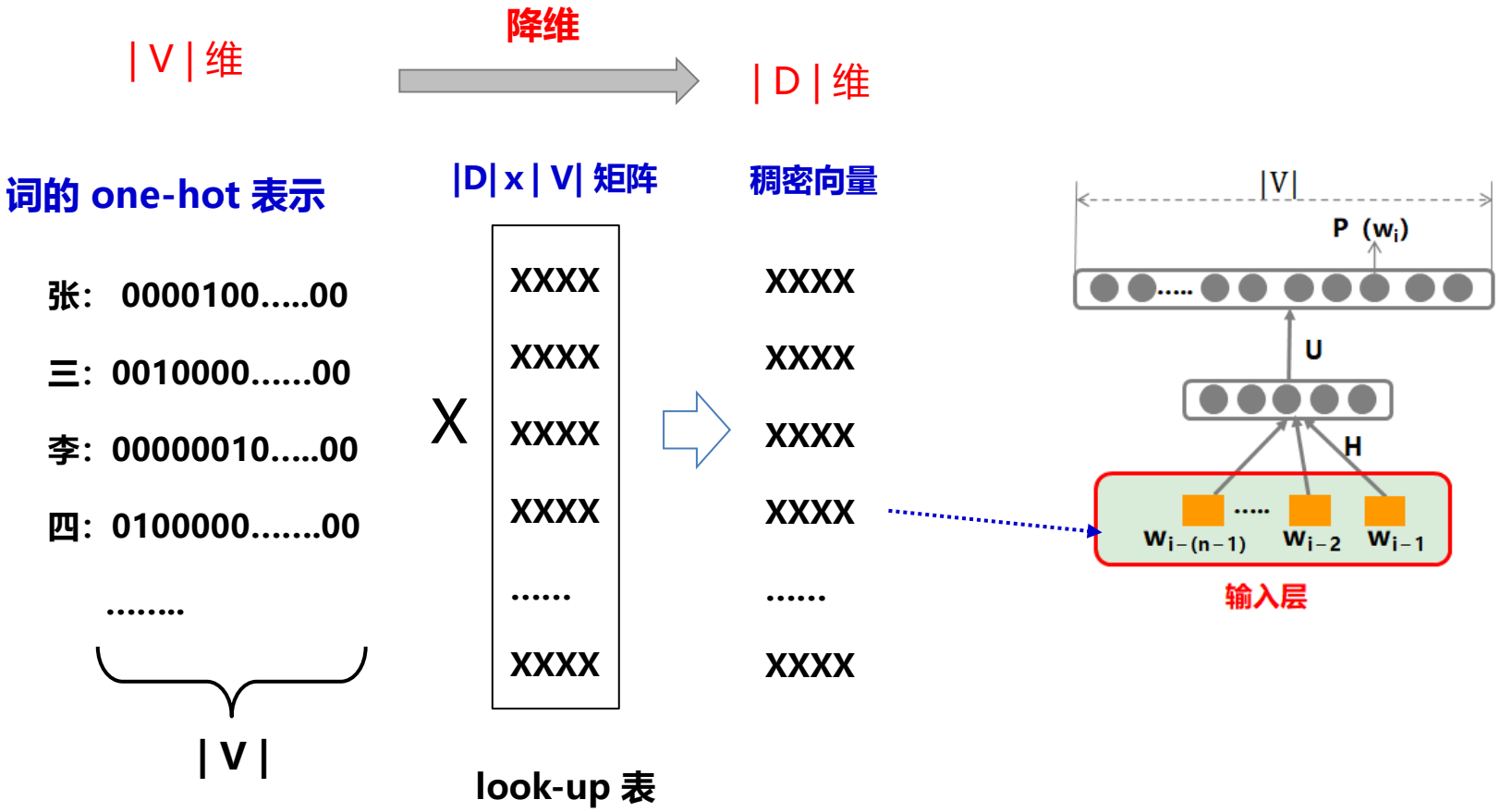
四：0100000.....00

.....

|V|

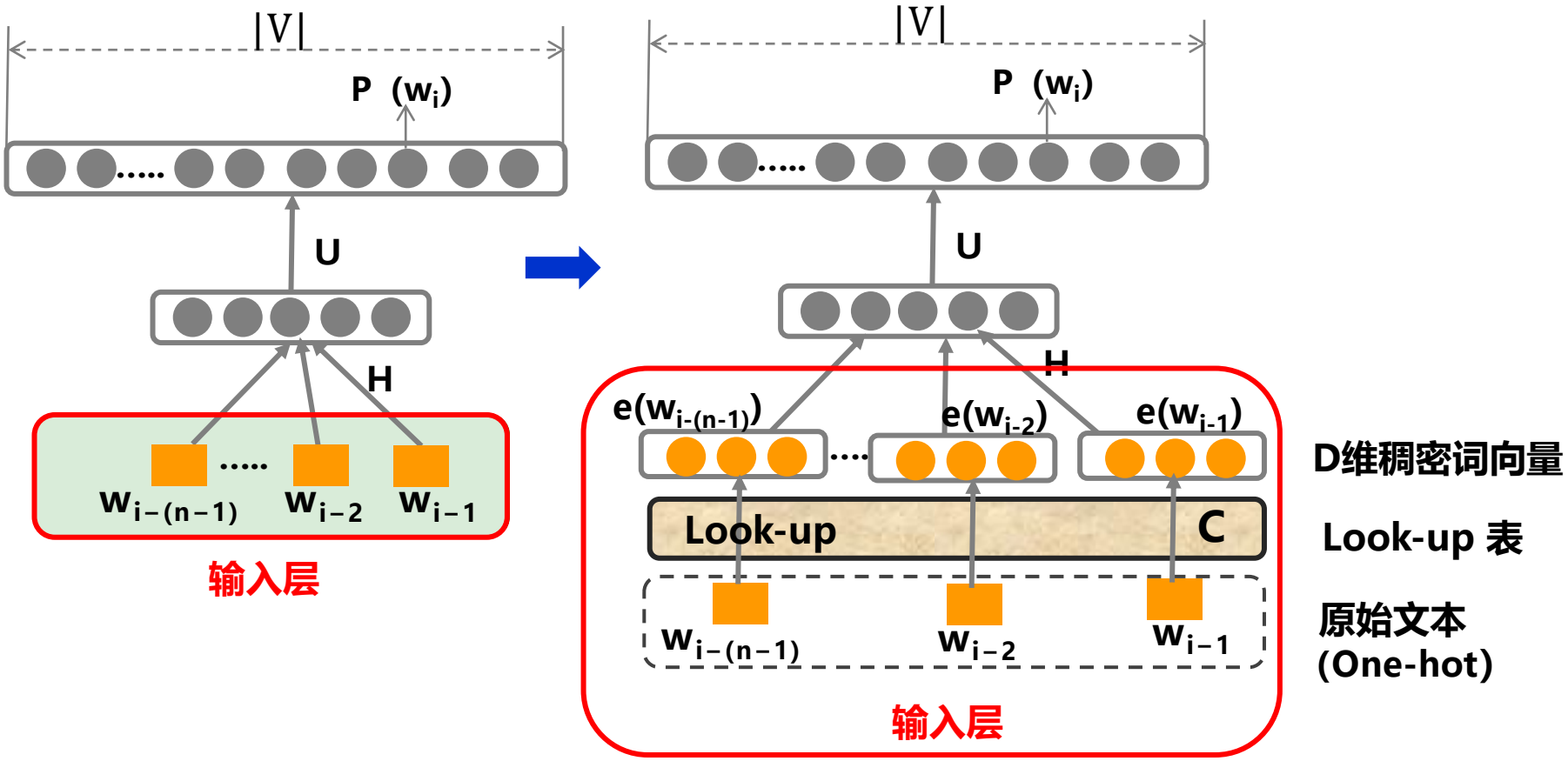
Yoshua Bengio, et.al. A neural probabilistic language model（2001, 2003）

1. NNLM模型(词向量)


| V | 维
降维
| D | 维
词的 one-hot 表示
|D| x | V| 矩阵
稠密向量
张： 0000100.....00
三：0010000.....00
李：00000010.....00
四：0100000.....00
.....
| V |
X
XXXX
XXXX
XXXX
XXXX
.....
XXXX
XXXX
XXXX
XXXX
.....
XXXX
look-up 表
|V|
P (wi)
U
H
wi-(n-1)
wi-2
wi-1
输入层

1. NNLM模型(词向量)

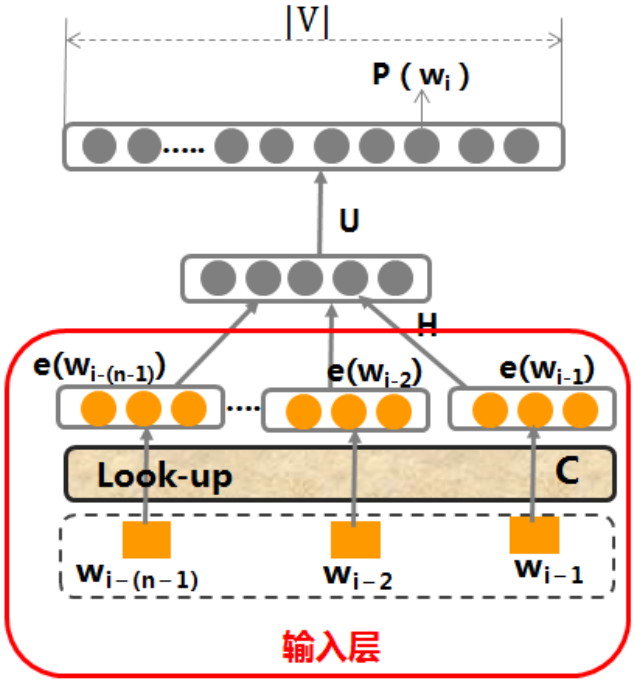
NNLM模型-输入表示



Yoshua Bengio, et.al. A neural probabilistic language model（2001, 2003）

1. NNLM模型(词向量)

NNLM模型-输入表示



D维稠密词向量

Look-up 表

原始文本
(One-hot)

例，look-up 表 C 如下：

$$C = \begin{pmatrix} (w_1)_1 & (w_2)_1 & \cdots & (w_V)_1 \\ (w_1)_2 & (w_2)_2 & \cdots & (w_V)_2 \\ \vdots & \vdots & \ddots & \vdots \\ (w_1)_D & (w_2)_D & \cdots & (w_V)_D \end{pmatrix}$$

w_2 = [0　　1　　0... 0]

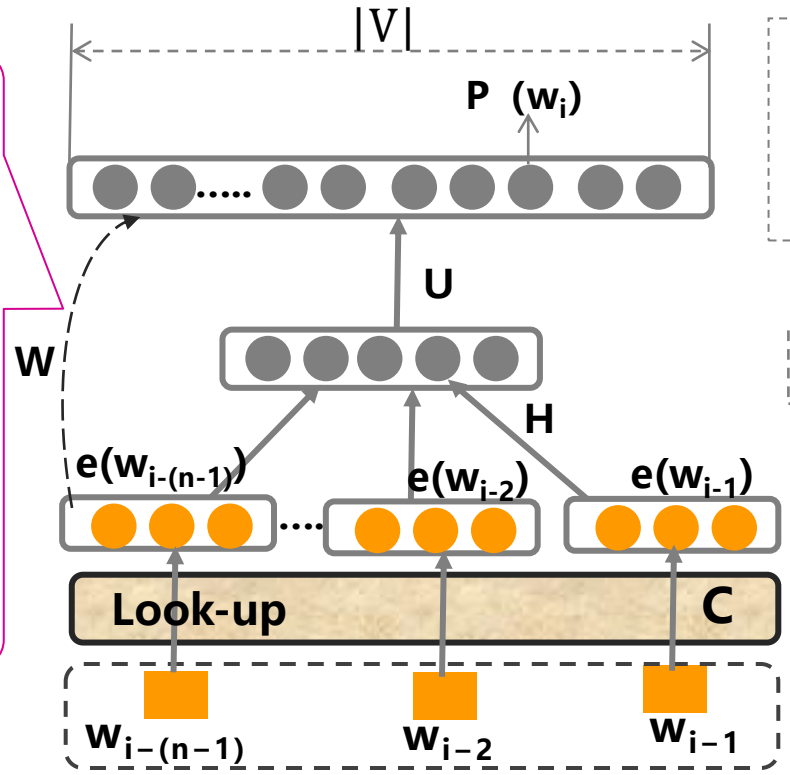
$e(w_2) = (w_2)_1\ (w_2)_2\ ....(w_2)_D$

稠密向量表示**Look-up**表 **C** 是|D| x | V|维实数投影矩阵，| V|表示词表的大小，|D| 表示词向量e的维度（一般50维以上）；各词的词向量存于C中。词w 到其词向量e(w) 的转化是从该矩阵中取出相应的列。

Yoshua Bengio , et.al. A neural probabilistic language model（2001, 2003）

1. NNLM模型(词向量)

(1) NNLM模型结构(词向量)



用该直连边边，迭代次数可减半；不用生成语言模型性能更好。后续工作，很少有使用该边

softmax(y)

输出层：$p(w_i|w_{i-(n-1)} \dots w_{i-1}) = \frac{\exp(y(w_i))}{\sum_{k=1}^{|V|}\exp(y(v_k))}$

$y(w_i) = b^2 + Wx + U h$

隐藏层：$h = \tanh(XH + b^1)$

输入层：X: n-1个词 $w_{i-(n-1)}, \dots, w_{i-1}$ (词向量初值)

的词向量拼接 $X = [e(w_{i-(n-1)}) \dots e(w_{i-1})]$

参数：$\theta = \{H, U, W, b^1, b^2,$ 词向量 $\}$

训练结束→训练好的词向量

Yoshua Bengio , et.al. A neural probabilistic language model（2001, 2003）

1. NNLM模型(词向量)

(2) NNLM模型(词向量)学习

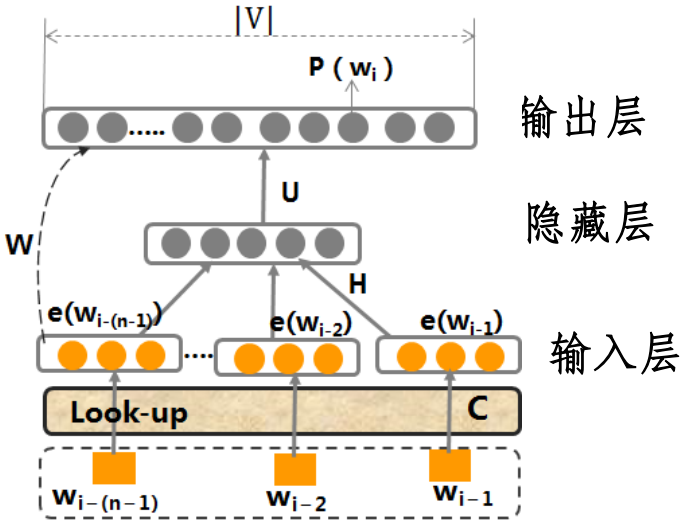
参数： $\theta = \{H, U, W, b^1, b^2, \text{词向量}\}$

输出： $p(w_i \mid w_{i-(n-1)} \cdots w_{i-1})$

$$= \frac{\exp(y(w_i))}{\sum_{k=1}^{|V|} \exp(y(v_k))}$$

$y(w_i) = b^2 + Wx + U(\tanh(XH + b^1))$

$X = [e(w_{i-(n-1)}) \ldots e(w_{i-1})]$



- **语料：（“无监督”）**

文本： $S = w_1, w_2, \cdots, w_n, \cdots$

实例： X： $w_1, w_2, \cdots, w_{i-1}$

$\hat{Y}$ ： w_i

- **目标函数：**

采用log损失函数 $L(Y, P(Y|X)) = -\log P(Y|X)$

对于整个语料而言，语言模型需要最大化：

$$\sum_{w_{i-(n-1)}} \sum_{i \in D} \log P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

- **参数训练：**

(BP) 随机梯度下降法优化训练目标:

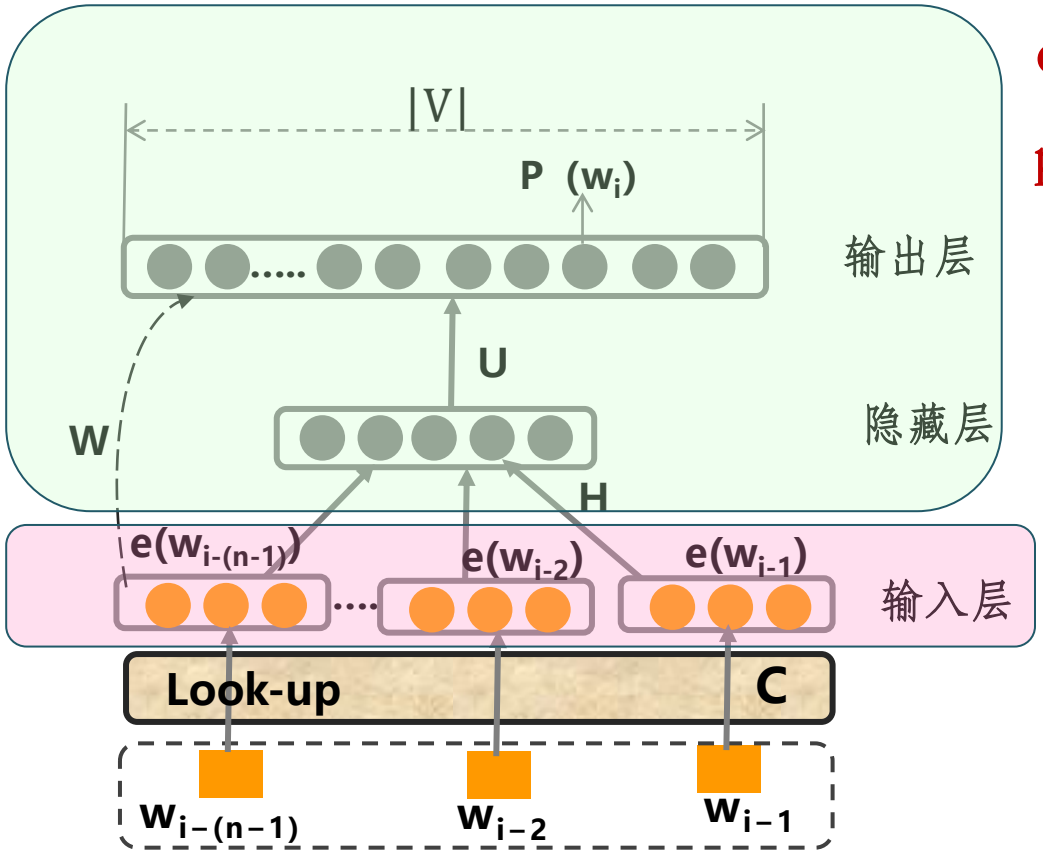
每次迭代，随机从语料D 中选取一段文本 $w_{i-(n-1)}, \cdots, w_i$ 作为训练样本进行一次梯度迭代

$$\theta \leftarrow \theta + \alpha \frac{\partial \log P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})}{\partial \theta}$$

其中，α 学习率，$\theta = \{H, U, W, b^1, b^2, \text{词向量}\}$

1. NNLM模型(词向量)

(3) NNLM模型作用



- 语言模型

$p(w_i | w_{i-(n-1)} \dots w_{i-1})$

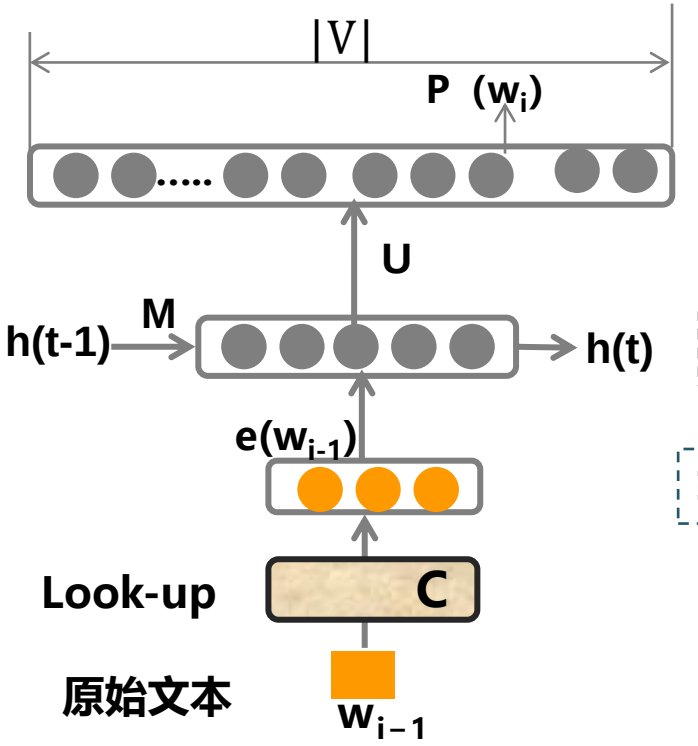
- 词向量

$w_{i-(n-1)} \dots w_{i-1}$ 的词向量

Yoshua Bengio , et.al. A neural probabilistic language model（2001, 2003）

2. RNNLM模型(词向量)

(1) RNNLM模型结构(词向量)



softmax(y)

输出层: $p(w_i|w_{i-1}) = \frac{\exp(y(w_i))}{\sum_{k=1}^{|V|}\exp(y(v_k))}$

$y(w_i)=b^2+U\,h(t)$

隐藏层: $h(t)=\tanh(X+Mh(t-1)+b^1)$

输入层: X: 词 w_{i-1}的词向量 $X= e(w_{i-1})$

输入: $X= e(w_{i-1})$ (词向量初值)，$h(t-1)$

输出: $p(w_i|w_{i-1})$ ， $h(t)$:(内部隐层)

参数: $\theta=\{U, M, b^1, b^2,$ 词向量$\}$

训练结束→训练好的词向量

- Tomas Mikolov, et.al.Statistical language models based on neural networks.2012
- Tomas Mikolov, et.al.Recurrent neural network based language model.2010

2. RNNLM模型(词向量)

(2) RNNLM模型(词向量)学习

参数： $\theta = \{U, M, b^1, b^2, \text{词向量}\}$

输出： $p(w_i \mid w_{i-1}) = \dfrac{\exp(y(w_i))}{\sum_{k=1}^{|V|} \exp(y(v_k))}$

$y(w_i) = b^2 + U(\tanh(XH + Mh(t-1) + b^1))$

- **语料：（“无监督”）**

文本： $S = w_1, w_2, \ldots, w_n$,

……

实例： X： START, $w_1, w_2, \ldots, w_{n-1}$

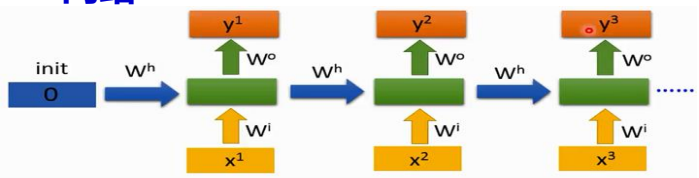
$\hat{Y}$ ： $w_1, w_2, \ldots, w_{n-1}\ w_n$

- **目标函数：**

对于整个语料，语言模型需要最大化

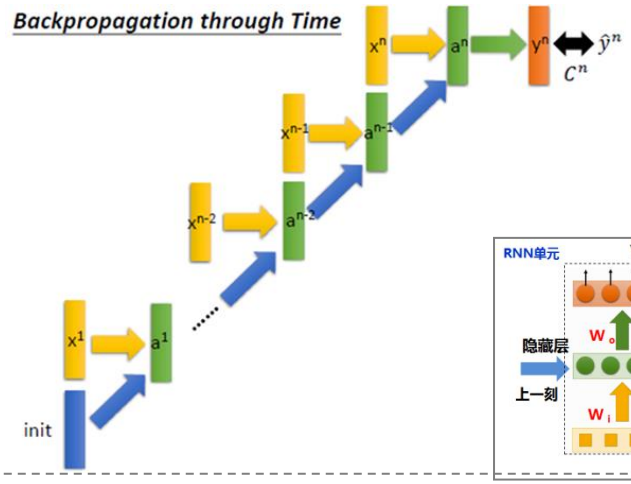
$$\sum_{w_{i-1}} \sum_{i \in D} \log P(w_i \mid w_{i-1})$$

RNN网络



- **参数训练：**

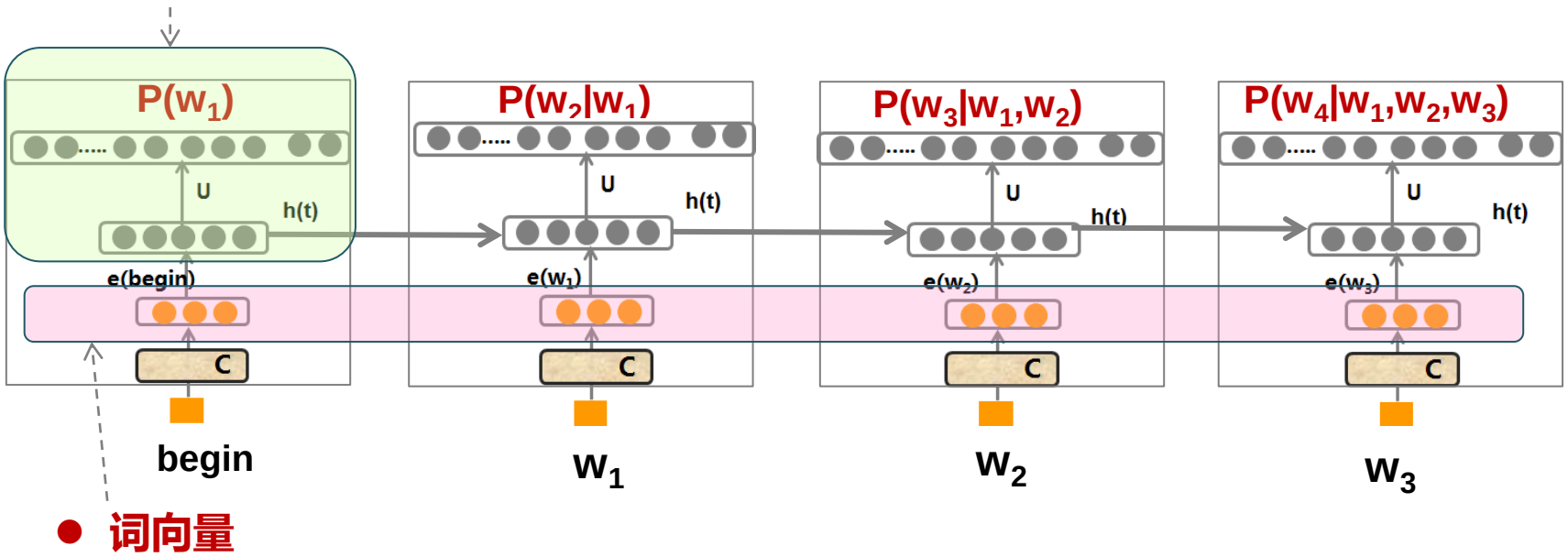
(BPTT) 随机梯度下降法优化训练目标:



2. RNNLM模型(词向量)

(3) RNNLM模型应用

- 语言模型



- 词向量

- Tomas Mikolov, et.al.Statistical language models based on neural networks.2012
- Tomas Mikolov, et.al.Recurrent neural network based language model.2010

3. C&W 模型

C&W 模型 （Collobert ,Weston ,2008）

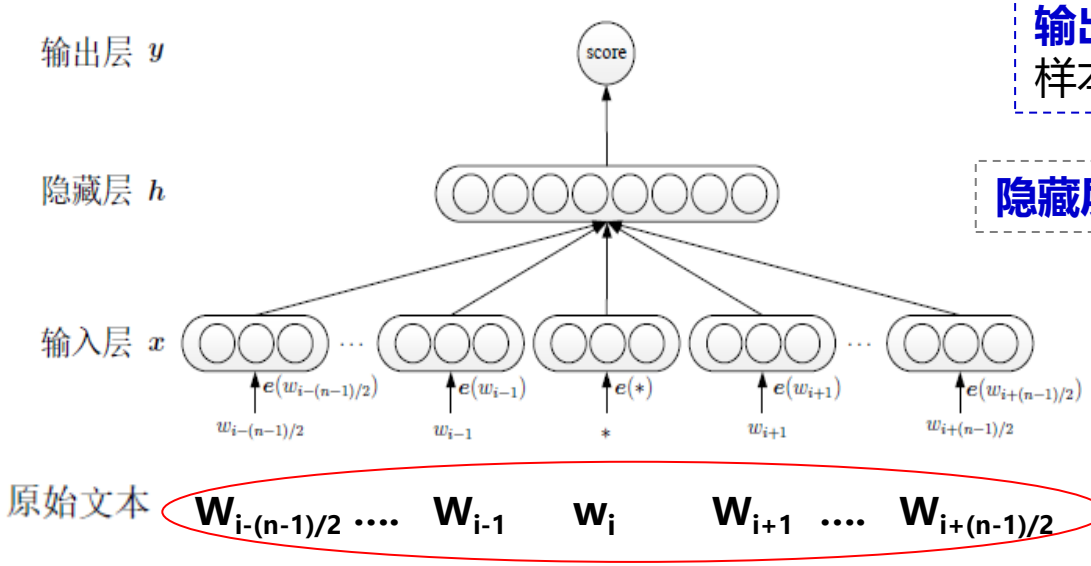
Collobe等人2008提出的第一个以直接快速生成词向量为目标的模型，采用直接对n元短语打分的方式替代语言模型中求解条件概率的方法：对于语料中出现过的n元短语，对其打高分；对于语料中没有出现的随机短语，对其打低分。通过这种方式，C&W模型可以更直接地学习得到符合分布假说的词向量。

特点：C&W 模型的目标函数是求目标词w与其上下文c的联合打分，而其他模型均为根据上下文c，预测目标词w。

Ronan Collobert : A unified architecture for natural language processing : Deep neural networks with multitask learning，2008

3. C&W 模型

(1) C&W模型结构



输出层：对输入序列打分（正样本打高分、负样本打低分）

隐藏层:上下文和目标词的联合表示

输入：目标词W_i及其上下文

$x = [e(w_{i-(n-1)}),..., e(w_i), ..., e(w_{i+(n-1)})]$

为从语料中选出的一个n元短语$w_{i-(n-1)/2}$，…, w_i, …, $w_{i+(n-1)/2}$一般n为奇数，以保证上文和下文的词数一致； w_i为目标词(序列中间的词) $x = [e(w_{i-(n-1)}),..., e(w_i), ..., e(w_{i+(n-1)})]$

Ronan Collobert:A unified architecture for natural language processing:Deep neural networks with multitask learning，2008

3. C&W 模型

(2) C&W模型学习

- **优化目标：**：

对于整个语料最小化:

$$\sum_{(w,c)\in D}\sum_{w'\in V}\max(0,1-\text{score}(w,c)+\text{score}(w',c))$$

其中，(w,c) :c 表示目标词 w 的上下文

- 正样本(w,c)来自语料
- 而负样本(w’ c)则是将正样本序列中的中间词替换成其它词

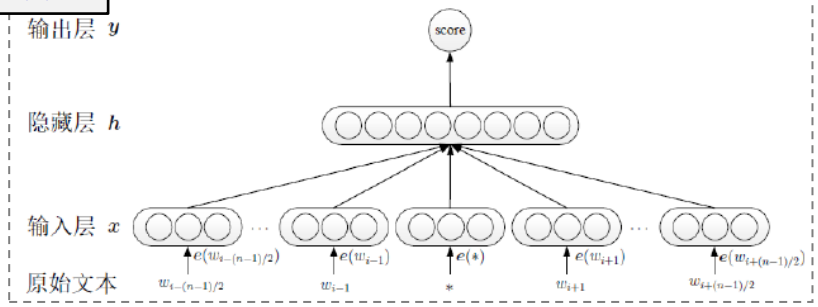
- **参数训练：**

采用pairwise的方式对文本片段进行优化，即可得词向量

3. C&W 模型

C&W模型与NNLN模型对比

C&W模型

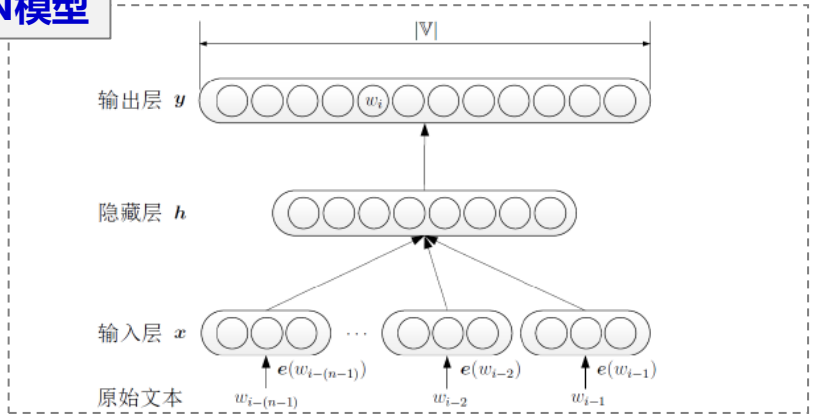


目标词在输入层

输出层只有1个节点

最后一层只需|h|次运算

NNLN模型



目标词在输出层

输出层有|V|个节点

最后一层需|Vl x lh|次运算，

且需要进行softmax运算

C&W 模型在运算速度上优于NNLM模型，但在许多语言学任务上，效果不如其它模型

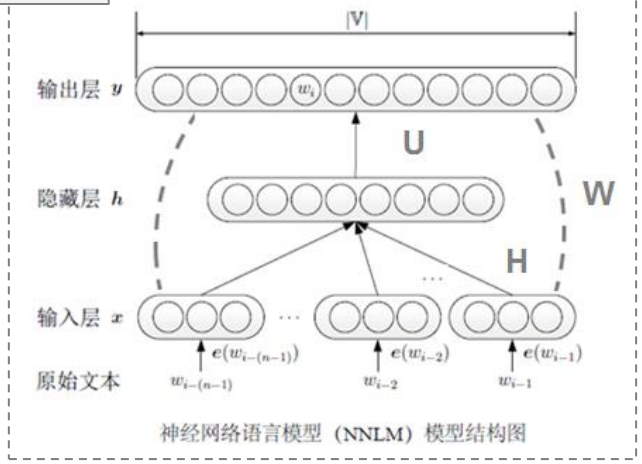
CBOW / Skip-gram模型

研表究明，汉字序顺并不定一影阅响读！事证实明了也许当你看这完句话之后才发字现都乱是的。

CBOW / Skip-gram模型

Mikolov等人在2013年，同时提出CBOW（Continuous Bag-of-Words）和Skip-gram模型。主要针对NNLM训练复杂度过高的问题进行改进，以更高效的方法获取词向量。

NNLN模型



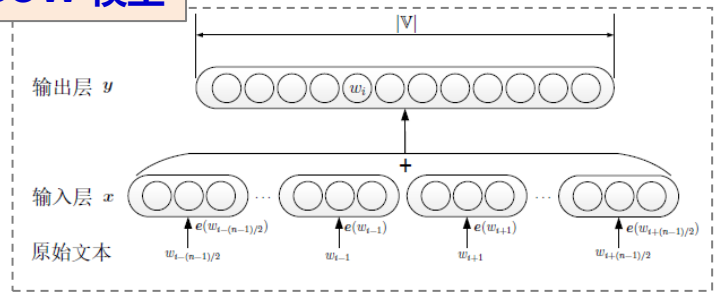
神经网络语言模型（NNLM）模型结构图

简化：

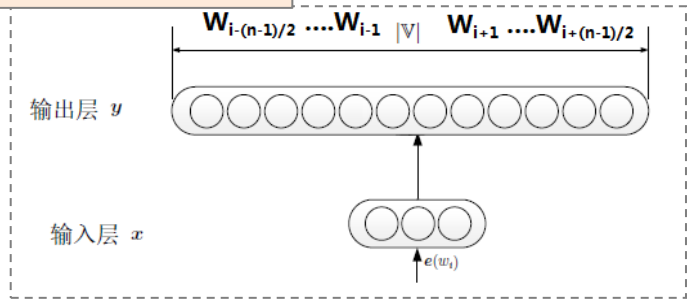
1. 除去隐藏层

2. 除去词序

CBOW 模型



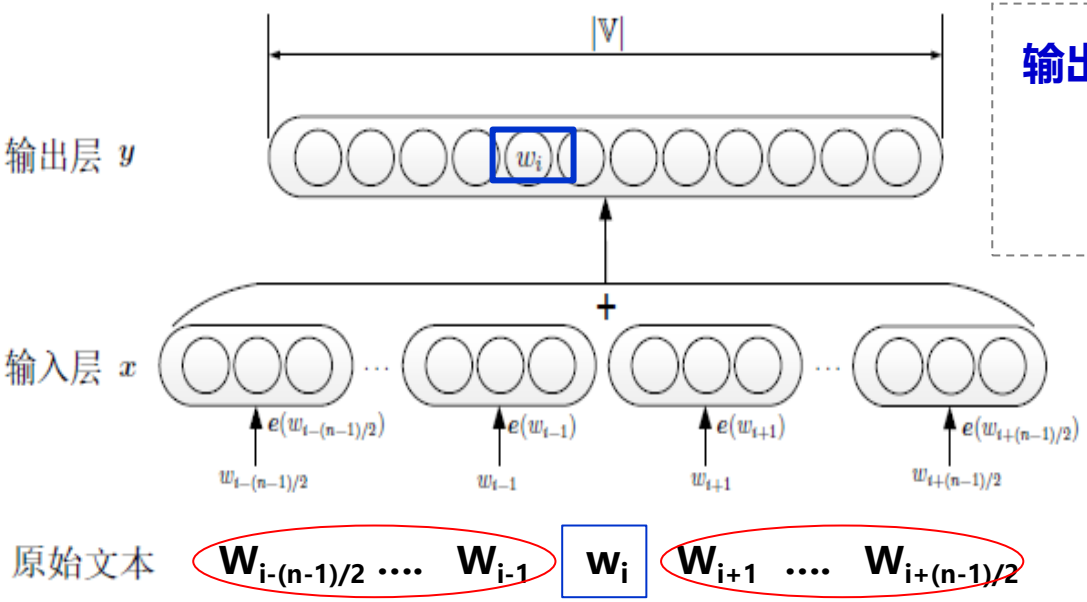
Skip-gram模型



在NNLM、RNNLM 和C&W 模型的基础上，简化模型，保留核心部分

4. CBOW 模型

■ CBOW 模型



输出层： $p(w_i|C) = \frac{\exp\left(e'(w_i)^{\mathrm{T}}x\right)}{\sum_{w'\in\mathbb{V}}\exp\left(e'(w')^{\mathrm{T}}x\right)}$

输入层： X: **词 w_i的上下文词向量平均值** $x = \frac{1}{n-1}\sum_{w_j\in c} e(w_j)$

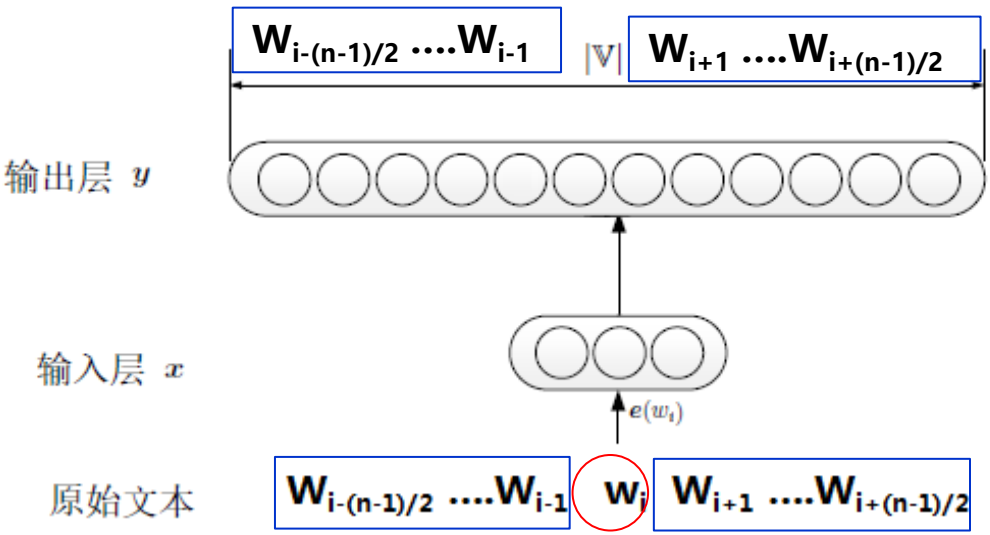
- **优化目标：** **最大化** $\sum_{(w,c)\in D}\log P(w|c)$
- **参数训练：** **梯度下降法**

为从语料中选出的一个n元短语$w_{i-(n-1)/2}$，…，w_i，…，$w_{i+(n-1)/2}$一般n为奇数，以保证上文和下文的词数一致；w_i为目标词，w_i上下文C为不包括w_i的n-1元短语

Tomas Mikolov:Efficient estimation of word representations in vector space，2013

5. Skip-gram模型

■ Skip-gram模型



输出层： $P(w_j \mid w_i)$

$$= \frac{\exp\left(e'(w_j)^{\mathrm{T}} x\right)}{\sum_{w' \in \mathrm{V}} \exp\left(e'(w')^{\mathrm{T}} x\right)}$$

输入层： X: 词 w_i 的词向量

- **优化目标：** 最大化 $\sum_{(w,c)\in D} \sum_{w_j \in c} \log P(w_j \mid w)$
- **参数训练：** 梯度下降法

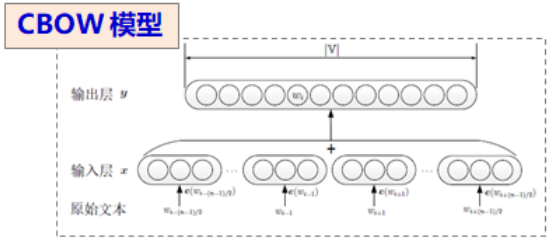
将目标词w_i的词向量作为输入，每次从w_i的上下文C 中选一个词作为预测词进行预测。目标词w_i及上下文C 定义同CBOW 模型

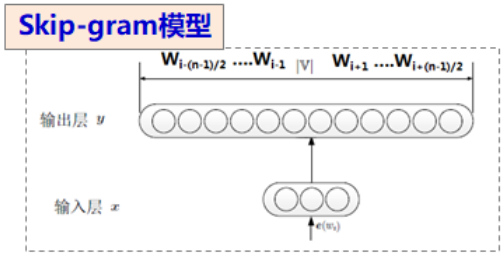
Tomas Mikolov:Efficient estimation of word representations in vector space，2013

CBOW / Skip-gram模型

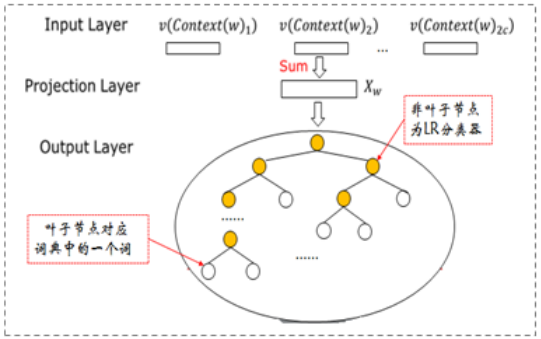
CBOW / Skip-gram模型中softmax优化问题：

计算 y 中的每个分量的值非常耗时，实际运算中采用层级softmax 函数优化。



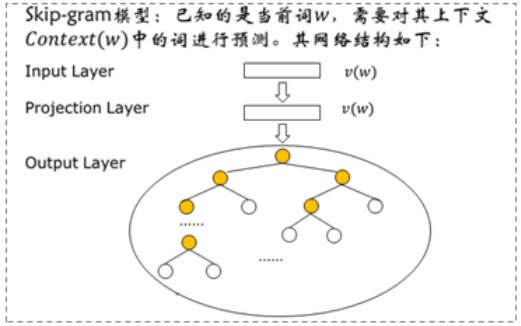


CBOW Hierarchical Softmax



求： $p(w_i|Context_i)$

Skip-gram Hierarchical Softmax



求： $p(Context_i \mid w_i)$

详情参阅：

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In Proceedings of the international workshop on artificial intelligence and statistics, , 2005.

CBOW / Skip-gram模型

基于CBOW/Skip-gram模型的词向量工具

Word2vec 是Google开源的将词表征为实数值向量的高效工具，其利用深度学习的思想，可以通过训练，把对词的处理简化为K维向量空间中的向量运算

Word2vec训练得到的词向量可以用于机器翻译，相似词查找，关系挖掘，中文聚类等任务中。

Word2vec总共有两种类型，每种类型有两个策略，总共4种

模型	CBOW		Skip-Gram	
方法	Hierarchical Softmax	Negative Sampling	Hierarchical Softmax	Negative Sampling

经典模型小结

模型特点

模型	目标词与上下文位置(n-gram)	模型输入	模型输出	目标词与上下文词之间的关系
NNLM	(上文)(目标词)	上文词向量拼接	目标词概率	上文在输入层、目标词在输出层，优化预测关系
C&W	(上文)(目标词)(下文)	上下文及目标词词向量拼接	上下文及目标词联合打分	上下文和目标词都在输入层，优化组合关系
CBOW	(上文)(目标词)(下文)	上下文各词词向量平均值	目标词概率	上下文在输入层、目标词在输出层，优化预测关系
Skip-gram	(上文)(目标词)(下文)	目标词词向量	上下文词概率	目标词在输入层、上下文在输出层，优化预测关系

经典模型小结

不同模型对比

指标	对比情况
模型复杂度	NNLM>C&W>CBOW>Skip-g ram
参数个数	NNLM>(cBoW=Skip-gram)>C&W
时间复杂度	NNLM>(cBoW=Skip-g ram)>C&W

内 容 提 要

9.1　词表示概述

9.2　浅层词表示模型

9.2.1　经典词表示模型

9.2.2　词向量特性及应用

9.2.2 词向量特性及应用

词向量具有如下语言学特性

- **语义相似的词，其词向量空间距离更相近 （分布假说）**

1-of-N Encoding

apple = [1 0 0 0 0]

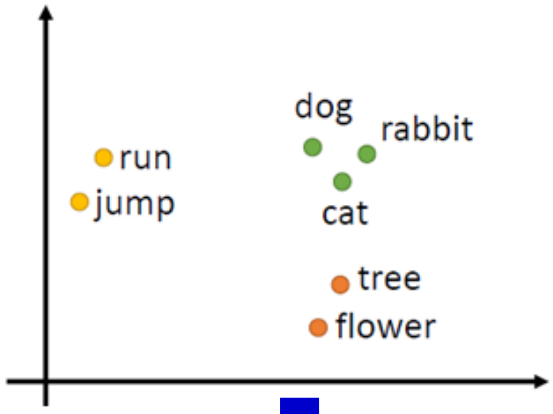
bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word Embedding

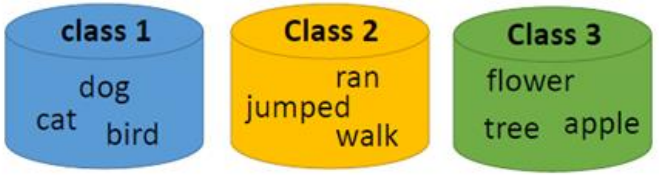


优点：降维，消除词汇鸿沟

其语言模型自带平滑功能

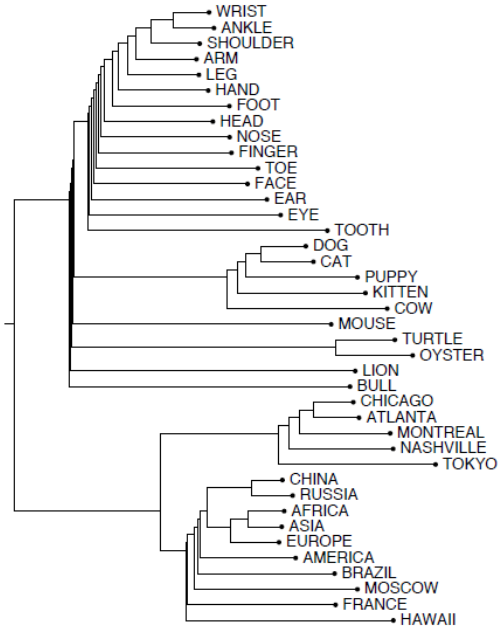
应用：同义词检测、单词类比等

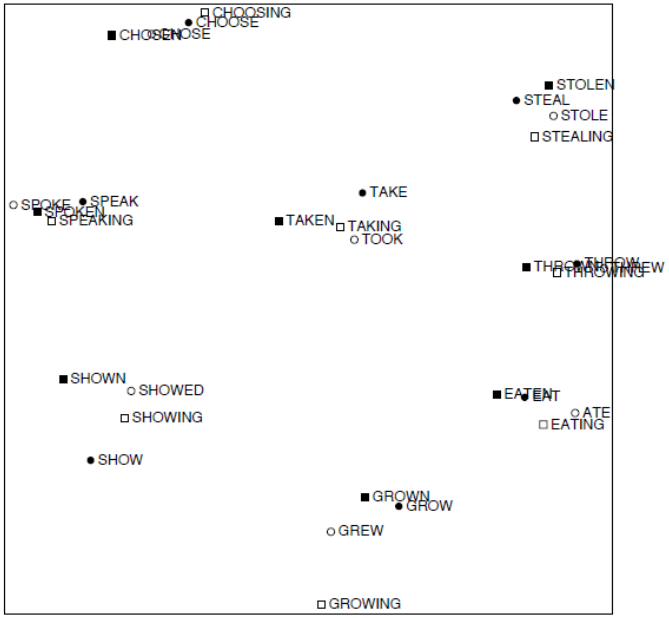
Word Class



9.2.2 词向量特性及应用

应用：语义相似度度量





Rohde et al. An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence, 2005

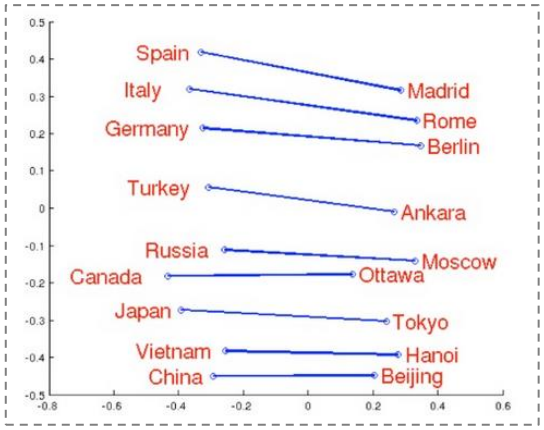
9.2.2 词向量特性及应用

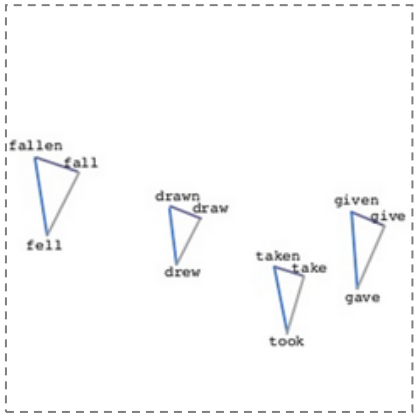
词向量具有如下语言学特性

- **相似关系词对的词向量之差也相似**

$$V(king) - V(queen) \approx V(uncle) - V(aunt)$$

$$V(hotter) - V(hot) \approx V(bigger) - V(big)$$



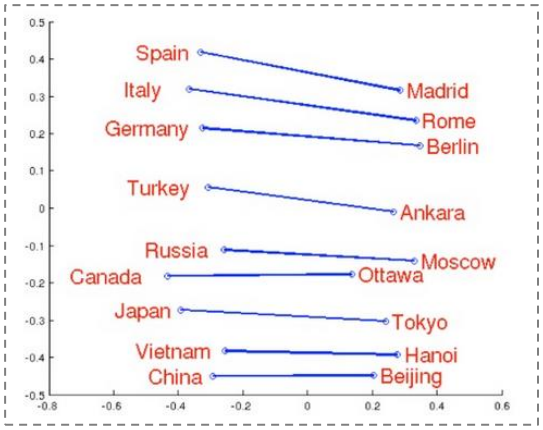


Source: http://www.slideshare.net/hustwj/cikm-keynotenov2014

9.2.2 词向量特性及应用

应用：直接使用词向量的加减法进行推理

如： Rome : Italy = Berlin : ?



用相似关系词对的词向量之差也相似，

直接使用词向量的加减法

$$\text{Compute } V(Berlin) - V(Rome) + V(Italy)$$

$$\text{Find the word w with the closest V(w)}$$

$$V(Germany) \approx \text{V(Berlin)} - \text{V(Rome)} + \text{V(Italy)}$$

Source: http://www.slideshare.net/hustwj/cikm-keynotenov2014

9.2.2 词向量特性及应用

词向量应用

- **利用词向量的语言学特性完成任务**

  利用词向量语义相似的词，其词向量空间距离相近的特征可完成语义相关性任务。如，同义词检测、单词类比等

- **将词向量作为静态特征（输入）**

使用词向量作为模型输入，在模型训练过程中，只调整模型参数，不调整输入词向量。如，基于平均词向量的文本分类、命名实体识别等

采用不同的词向量会影响系统性能。

- **将词向量作为动态初值（输入）（浅层表示模型）**

使用词向量作为神经网络的初始值，模型训练过程中会调整词向量的初值，从而提升神经网络模型的优化效果。

如，基于卷积神经网络的文本分类、词性标注等

9.2.2 词向量特性及应用

第3章： 概率语言模型存在问题

- 由于参数数量问题需要对词 i 的历史简化 n-gram
- 需要数据平滑

神经网络 “RNN 语言模型 + 词向量” 可以解决以上问题

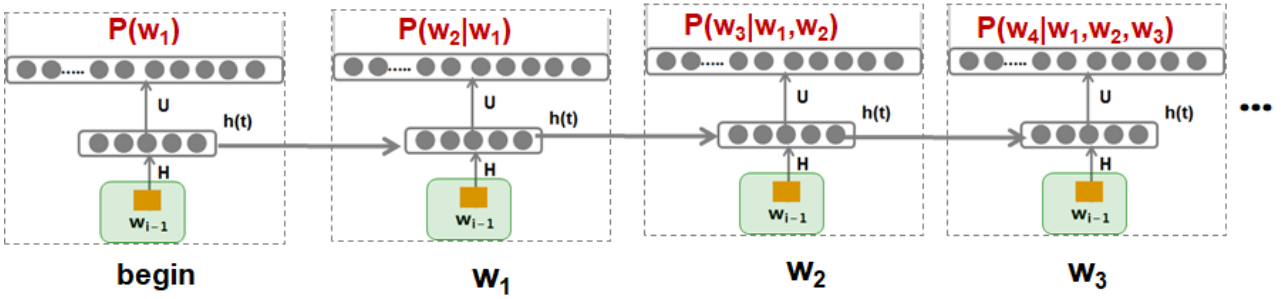
9.2.2 词向量特性及应用

问题1. 由于参数数量问题需要对词 i 的历史简化 n-gram

解决： **RNNLM模型**

$$P(w_1, w_2, w_3, \cdots, w_n)$$

$$= P(w_1)P(w_1 \mid w_2)P(w_3 \mid w_1, w_2) \cdots\cdots P(w_n \mid w_1, w_2 \cdots w_{n-1})$$



RNNLM模型可以保留每个词的所有历史信息

9.2.2 词向量特性及应用

问题2：需要数据平滑

平滑问题： $p(Cher\ read\ a\ book) = ?$

$$= p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

统计语言模型：

$$p(Cher|<BOS>) = \frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)} = \frac{0}{3}$$

$$p(read|Cher) = \frac{c(Cher\ \ read)}{\sum_w c(Cher\ \ w)} = \frac{0}{1}$$

<BOS>John read Moby Dick<EOS>

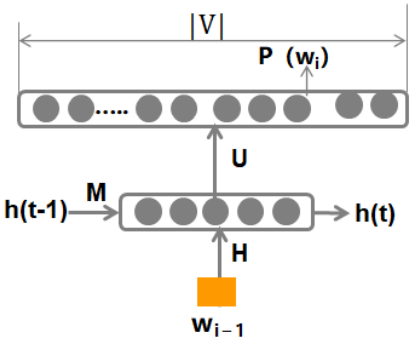
<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

$p(Cher\ read\ a\ book) = 0$ ➡ **需要数据平滑**

9.2.2 词向量特性及应用

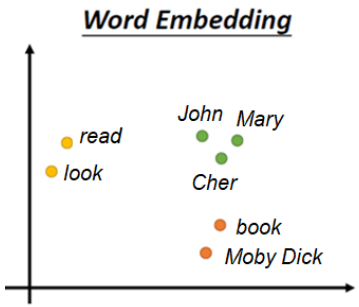
神经网络语言模型：



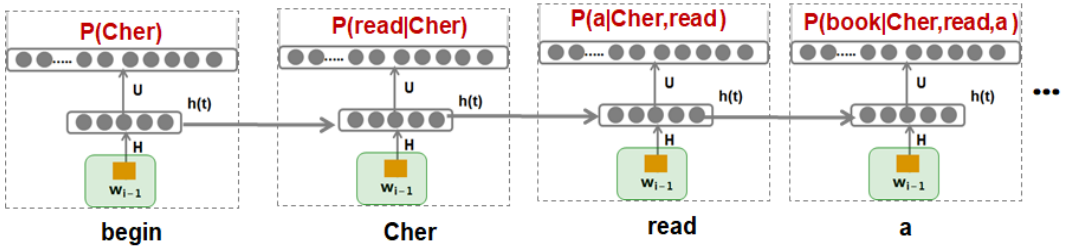
当语言模型训练好后，模型网络参数固定，

这时给任意的W_{i-1} $P(W_i)$ 不会为 0

词向量特征：



Cher 和 Join 的词向量比较接近



所以，采用预训练的词向量做输入，不需要数据平滑且效果好

参考文献：

李宏毅课程

http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML16.html

来斯惟，基于神经网络的词和文档语义向量表示方法研究，博士学位论文

licstar的博客： http://licstar.net/archives/328

深度学习word2vec学习笔记
http://download.csdn.net/detail/mytestmy/8565959

在 此 表 示 感 谢！

谢谢各位！

Q&A