

# 第5章 语言模型

(3/3)



# 本章内容

---

5.1 传统语言模型

5.2 神经语言模型

5.3 文本表示

5.3.1 动因

5.3.2 向量空间模型

 5.3.3 表示学习模型



## 5.3.3 表示学习模型

### ◆ 传统的文本语义概念表示模型

- 潜在语义分析 (latent semantic analysis, LSA)
- 主题模型 (topic model)  
(潜在狄利克雷分布, latent Dirichlet distribution (LDA))


### ◆ 基于深度学习的表示模型

通过深度学习模型以最优化某特定目标函数的方式，在分布式向量空间中学习文本的**低维实数向量表示**，即：

通过训练**将每个词映射成k维实数向量**（k为超参），然后通过向量间的距离来判断它们之间的语义相似度。



## 5.3.3 表示学习模型

- 词的表示学习  基于语言模型学习
    - CBOW and Skip-gram Model
    - BERT(Bidirectional Encoder Representations from Transformers)
    - ELMo (Embeddings from Language Models)
  - 短语表示学习
  - 句子表示学习
- word embedding  
Word2Vector
- 直接学习方法
    - C&W Model

## 5.3.3 表示学习模型

### ■ 基于语言模型的词向量学习

前馈神经网络语言模型训练过程

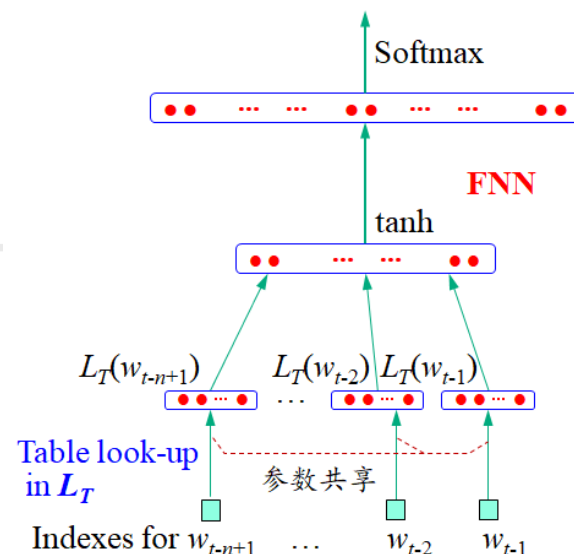
第一步：利用**查找表**（Lookup table）获得每个词的分布式表示（即词向量）。

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} W11 & W12 & W13 \\ W21 & W22 & W23 \\ W31 & W32 & W33 \\ W41 & W42 & W43 \\ W51 & W52 & W53 \end{bmatrix} = \begin{bmatrix} W41 & W42 & W43 \end{bmatrix}$$

## 5.3.3 表示学习模型

### ■ 基于语言模型的词向量学习

前馈神经网络语言模型训练过程



第二步：将表示context的n个词的词嵌入拼接（或累加）起来，通过一个隐藏层和一个输出层，最后通过softmax输出当前的 $p(w_t|\text{context})$ 。

之前我们假定查找表是给定的，事实上，该矩阵和网络参数  $W$ ,  $b$  一样，都是通过模型训练求解的。

也就是说：每种语言模型在训练过程中，都有一个副产品（词向量矩阵 **Lookup table**）



## 5.3.3 表示学习模型

---

### ■ CBOW(Continuous Bag-of-Words Model)

- ✧ 用某个词左右两边的词辅助预测当前词
- ✧ 词序不影响预测

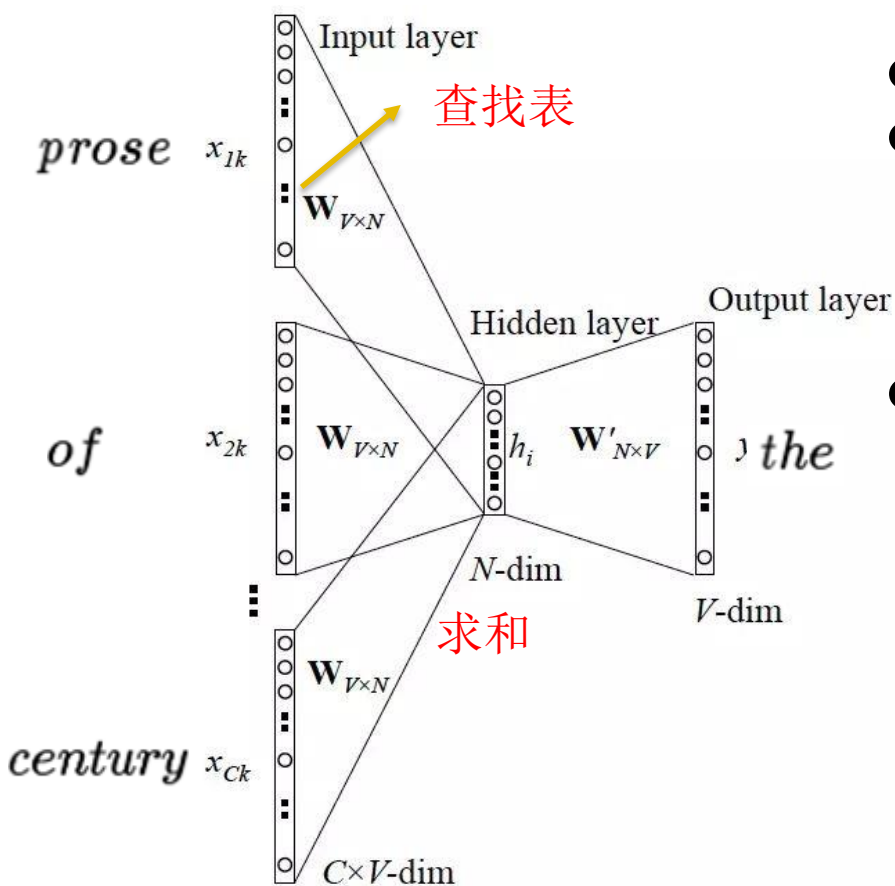
the florid prose of **the** nineteenth century

$P(\text{" the " } | (\text{" prose "}, \text{" of "}, \text{" nineteenth "}, \text{" century "}))$

## 5.3.3 表示学习模型

### CBOW

$P(\text{" the " } | (\text{" prose "}, \text{" of "}, \text{" nineteenth "}, \text{" century "}))$



- 输入：上下文单词 $x_i$ 的one-hot向量.
- 通过累加求平均得到隐层向量 $h$

$$h = \frac{1}{C} W \cdot \left( \sum_{i=1}^C x_i \right)$$

- 计算在输出层每个结点的值

$$u_j = v_{wj}'^T \cdot h$$

其中 $v_{wj}'^T$ 是输出矩阵 $W'$ 的第 $j$ 列。

$$p(w_j | w_1) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^V \exp(u_{j'})}$$



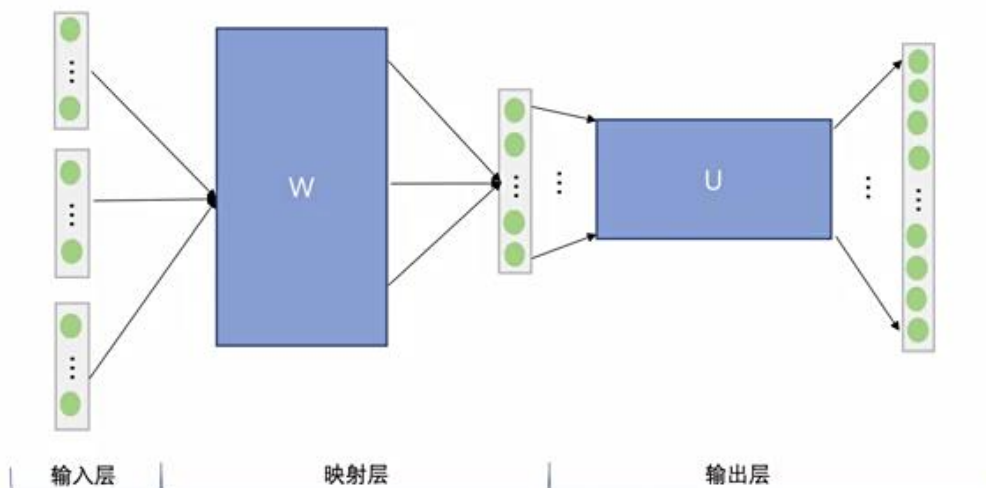
## 5.3.3 表示学习模型

### ■ CBOW(Continuous Bag-of-Words Model)

词典={我, 喜欢, 到处, 旅游}

句子=我 喜欢 到处 旅游

知乎

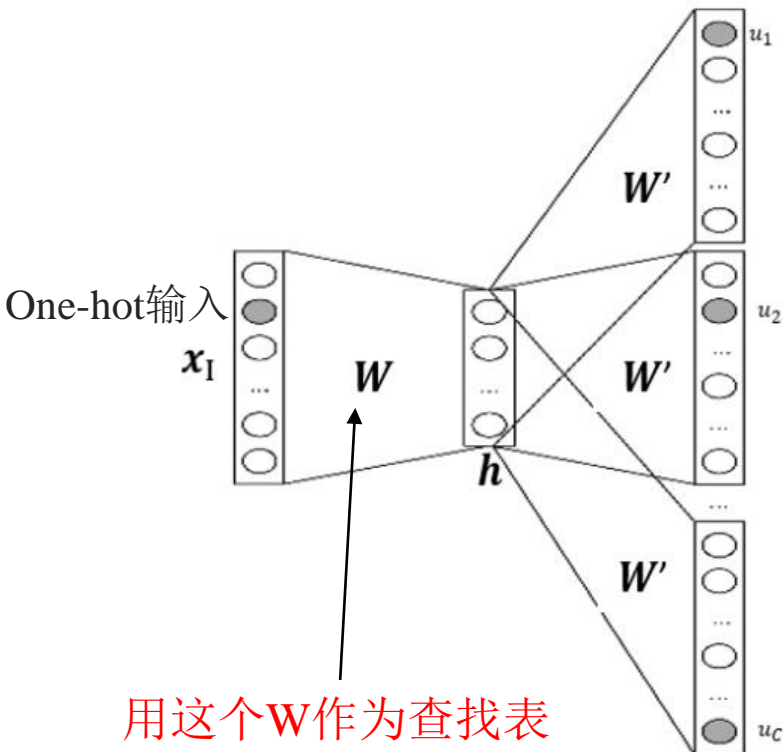


Step1.得到上下文及输出的one-hot向量

公众号: IT民工boby

## 5.3.3 表示学习模型

### • Skip-gram Model



skip-gram模型的输入是一个单词  $w_I$

它的输出是 $w_I$ 的上下文 $w_{O,1}, \dots, w_{O,C}$

如: I drive my **car** to the store。

如把” car” 作为输入, 窗口为4,  
词组{ “drive”, “my”, “to”, “the” }  
就是输出。

语言模型训练目标:

最大化  $P(\text{I drive my to the store} | \text{car})$

Lookup table  $W$ 仍然是语言模型的副产品



## 5.3.3 表示学习模型

---

### • C&W Model

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, *Proc. ICML'2008*

#### ✧ 基本思路

由上下文词预测当前词，使其概率最大。

C&W 模型的目标是生成词向量，而skip-gram和CBOW模型的目标是生成语言模型  $P(w_i | w_1, w_2, \dots, w_{i-1})$ 。

因此，C&W模型采用了一种更高效的方法，直接对  $n$  元短语打分的训练方式。区别于CBOW模型，C&W模型中 $w_i$ 在输入而非输出端。

## 5.3.3 表示学习模型

✧ 举例说明

$$(w_i, Context) = w_{i-n}, \dots, w_{i-1}, \mathbf{w}_i, w_{i+1}, \dots, w_{i+n}$$

we have learned a **lot** from this lesson



随机替换

$$(w'_i, Context) = w_{i-n}, \dots, w_{i-1}, \mathbf{w}'_i, w_{i+1}, \dots, w_{i+n}$$

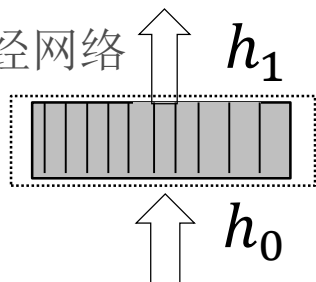
we have learned a **today** from this lesson

$$score(w_i, Context) > score(w'_i, Context)$$

## 5.3.3 表示学习模型

Right / Random

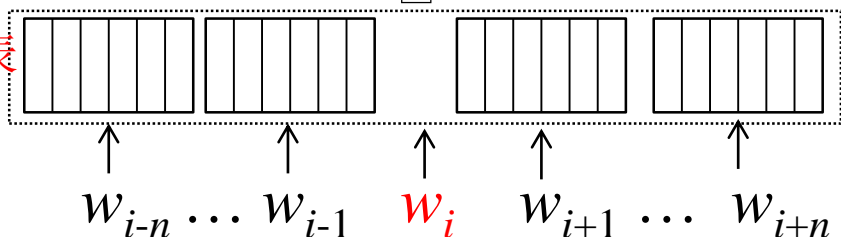
简单的前馈神经网络



$$h_1 = f(W_0 h_0 + b_0)$$

$$\text{score}(w_i, \text{Context}) = W_1 h_1 + b_1$$

查表



$$h'_1 = f(W_0 h'_0 + b_0)$$

$$\text{score}(w'_i, \text{Context}) = W_1 h'_1 + b_1$$

one-hot输入

$$h_0 = [e(w_{i-n}), \dots, e(w_{i-1}), e(w_i), e(w_{i+1}), \dots, e(w_{i+n})]$$

$$h'_0 = [e(w_{i-n}), \dots, e(w_{i-1}), e(w'_i), e(w_{i+1}), \dots, e(w_{i+n})]$$

注意：h中的e() 可以理解为查表操作，获取到每个w<sub>i</sub>的词向量

## 5.3.3 表示学习模型

$$score(w_i, Context) > score(w'_i, Context) + 1$$

希望每一个正样本应该比对应的负样本打分高1分

$$0 > score(w'_i, Context) + 1 - score(w_i, Context)$$

对下列目标函数进行最小化优化

$$loss = \sum_{(w_i, C) \in D} \sum_{w' \in V'} \max(0, \underline{1 + score(w'_i, Context) - score(w_i, Context)})$$

✧ 问题：上下文中的词顺序对预测结果有直接的影响。



# 部分开源的词向量学习工具

---

- Google Word2Vec  
<http://code.google.com/p/word2vec/>
- EMLo: 基于循环神经网络预训练模型  
<https://github.com/allenai/bilm-tf>
- GPT: 基于单向自我注意机制的预训练语言模型(Language Model with Generative Pre-training)  
<https://github.com/openai/gpt-2>
- **BERT**: 基于双向自我注意机制语言模型(Bidirectional Encoder Representations from Transformer)  
<https://github.com/google-research/bert>
- .....



# 本部分小结

---

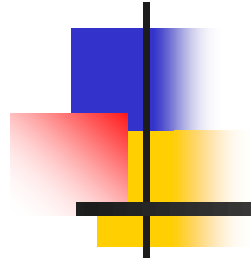
## ◆ 向量空间模型

特征项 与 特征项权重

## ◆ 表示学习模型

- 词的表示学习





***Thanks***

