

语言模型

杨沐昀

语言技术研究中心
哈尔滨工业大学

目录

- * 语言模型的概念
- * 分词：从规则到统计的模型发展
- * n元语言模型
 - * n元文法
 - * 最大似然估计
 - * 语言模型性能评价
 - * 平滑
- * 神经网络语言模型
 - * 前馈神经网络语言模型
 - * 循环神经网络语言模型

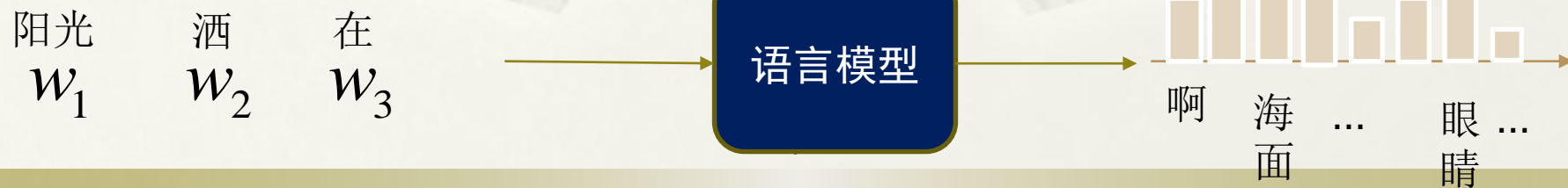
语言模型

- * 字面上：语言的数学建模
 - * 语言整体系统 还是 语言具体层次
- * 领域内：语言模型是用来描述自然语言词汇分布的模型，包括统计语言模型和神经网络语言模型
 - * 最初应该是源自语音识别领域里
- * 利用语言模型，也可以在给定上文条件下对接下来可能出现的词进行预测



语言模型

- * 统计语言模型：用来描述自然语言概率分布的模型
- * 给定一个词列 $w_1w_2w_3...w_n$ ，语言模型根据习得的统计规律给出这个词序列作为一句话被产生的概率 $p(w_1w_2w_3...w_n)$
- * 这一个技术引用范围广、效果好。



目录

- * 语言模型的概念
- * 分词：从规则到统计的模型发展
- * n元语言模型
 - * n元文法
 - * 最大似然估计
 - * 语言模型性能评价
 - * 平滑
- * 神经网络语言模型
 - * 前馈神经网络语言模型
 - * 循环神经网络语言模型

分词的提出和定义

- * 词：是自然语言中能够独立运用的最小单位，是语言信息处理的基本单位。

- * わたしはとうきょうだいがくりゅうがくせいです。

- * سيارة دمرت بقصف أميركي لمدينة الصدر وفي الإطار عزت الدوري

- * 分词：将句子转换成词序列

- * 东亚语言比较突出：中文、日文、韩语等

分词的意义

- * 自动分词是正确的中文信息处理的基础

- * 文本检索

- * 和服 | 务 | 于三日后裁制完毕，并呈送将军府中。
 - * 王府饭店的设施 | 租 | 服务 | 是一流的。
如果不分词或者“和服务”分词有误，都会导致荒谬的检索结果。

- * 文语转换

- * 他们是来 | 查 | 金泰 | 撞人那件事的。（“查” 读音为cha）
 - * 行侠仗义的 | 查金泰 | 远近闻名。（“查” 读音为zha）

什么是词：分词困难（1）

- * 汉语词定义不明确
 - * 牛肉是词，马肉是不是？
 - * 打倒是词，打死、打伤、饿死、涂黑是不是？
- * 采用“分词单位”的说法，建立词表
 - * 取舍理由不够充分，人为色彩过重
 - * 过于复杂，难于把握
- * 为操作的方便，必须确定统一的标准或规范

分词标准

* 汉语分词规范问题的提出

- * 分词是许多技术的基础：语音识别、信息检索、机器翻译等
- * 中文词之间没有明显分界符，不同人对同一句话词的界限有不同的看法，需要一个同一的标准。
- * 863/973和SIGHAN对计算机分词结果的评价都以人工分词结果作为标准，人工结果是否科学规范？

* 公布的规范

- * 《信息处理用现代汉语分词规范，中华人民共和国国家标准（GB/T13715）》
- * 《北京大学现代汉语语料库基本加工规范，北京大学，2002》
- * 《现代汉语语料库文本分词规范（Ver3.0），北京语言文化大学语言信息处理研究所、清华大学计算机科学与技术系，1998.12.09》
- * 《973当代汉语文本语料库分词、此行标注加工规范，山西大学，2003》
- * 《咨询处理用中文分词规范，台湾省，1998》

分词算法

- * 理性主义

- * 正向最大匹配分词

- * 逆向最大匹配分词

- * 双向最大匹配分词

- * 最短路径分词

- * 经验主义

- * 最大词频分词

理性主义的分词方法

- * 使用预先建立的词典（根据分词规范得到的词表）
- * 依赖人的语言观察和经验直觉设计算法
 - * 长度和频率
- * 从研究角度来看，启发式函数设计过于主观
 - * 假设条件过强
 - * 并未建立与问题本质的联系
 - * 均属于贪心策略，未及考虑全局最优

正、反向最大匹配分词


- * 正向最大匹配(FMM)：从左往右，每次匹配词典中最长的词条
 - * “市场/中国/有/企业/才能/发展/”
 - * 错误切分率为1 / 169
- * 正向最大匹配(BMM)：从右往左...
 - * “市场/中/国有/企业/才能/发展/”
 - * 错误切分率为1 / 245(更有效)
- * 双向最大匹配：用于发现分词歧义
 - * “市场/ (中国有) /企业/才能/发展/”

最大匹配法的问题

- * 存在分词错误：增加知识、局部修改
- * 局部修改：增加歧义词表，排歧规则

规则示例

IF $W = \text{"个人"}, W_{\text{Left}} = \text{数词}$ THEN $W = \text{"个/ 人/"}$ ENDIF



歧义词表
...
才能
个人
家人
马上
研究所
...

最少分词法—对最大匹配的算法优化

- * 分词结果中含词数最少

- * 等价于最短路径

- * 好于单向的最大匹配方法

- * 最大匹配：独立自主/和平/等/互利/的/原则

- * 最短路径：独立自主/和/平等互利/的/原则

- * 实现方法

- * 动态规划算法

最大词频分词法：经验主义登场

- * 基本思想：出现频率越高的词越可靠
 - * 正确率可达到92%*（和FMM实验条件不一样）
 - * 简便易行，效果一般好于基于词表的方法

分词歧义:分词难题(2)

* 交集型切分歧义

- * 汉字串AJB被称作交集型切分歧义，如果满足AJ、JB同时为词(A、J、B分别为汉字串)。此时汉字串J被称作交集串。

- * [例] “结合成分子”

- * 结合 | 成 分 | 子 |

- * 结合 | 成 | 分子 |

- * 结 | 合成 | 分子 |

- * [例] “**美国**会通过台售武法案”

- * [例] “**乒乓球**拍卖完了”

分词歧义：分词难题(2)

* 组合型切分歧义

- * 汉字串AB被称作组合型切分歧义，如果满足条件：A、B、AB同时为词

- * [例]组合型切分歧义：“起身”

- * 他站 | 起 | 身 | 来。

- * 他明天 | 起身 | 去北京。

* 歧义的挑战：链长(交集字段中含有交集字段的个数)

- * 链长为1：和尚未
- * 链长为2：结合成分
- * 链长为3：为人民工作
- * 链长为4：中国产品质量
- * 链长为5：鞭炮声响彻夜空
- * 链长为6：努力学习语法规则
- * 链长为7：中国企业主要求解决
- * 链长为8：治理解放大道路面积水

分词歧义的实际分布

交集型歧义：组合型歧义 = 1: 22 语料规模：17,547字 [1]

语料规模：500万字新闻语料 [2]

链长 歧义 字段	1	2	3	4	5	6	7	8	总计
Token次数	47402	28790	1217	608	29	19	2	1	78248
比例%	50.58	47.02	1.56	0.78	0.04	0.02	0.00	0.00	100
Type种数	12686	10131	743	324	22	5	2	1	23914
比例%	53.05	42.36	3.11	1.35	0.09	0.02	0.01	0.01	100

[1] 刘挺、王开铸，1998，关于歧义字段切分的思考与实验。《中文信息学报》第2期，63-64页。

[2] 刘开瑛，2000，《中文文本自动分词和标注》，商务印书馆，65页。

Type是token去重后的结果

■ 新词与未登录词：分词难题(3)

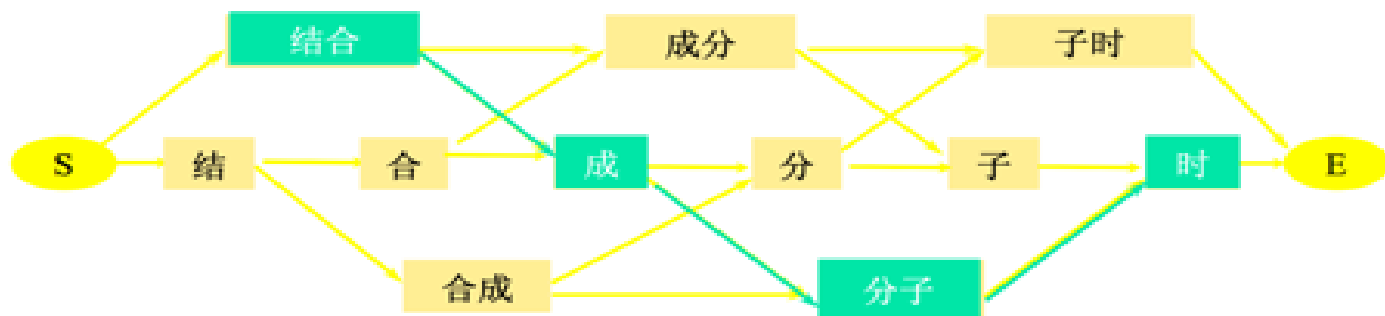
- * 虽然一般的词典都能覆盖大多数的词语，但有相当一部分的词语不可能穷尽地收入系统词典中，这些词语称为未登录词或新词

- * 分类：

- * 专有名词：中文人名、地名、机构名称、外国译名、时间词
- * 重叠词：“高高兴兴”、“研究研究”
- * 派生词：“一次性用品”
- * 与领域相关的术语：“互联网”

汉语分词：问题求解视角

- * 问题空间：（基于词表的）全切分有向图
- * 解：由S到E最优路径（动态规划搜索）
- * 核心挑战：路径评价函数
 - * 机器学习角度：损失函数



基于N元语法的分词

- * 基于N元语法的切分排歧

- * {Text: 输入文本, 可以是一句话}

为何这样开始, 直接用结果做假设求证可否?

$$\hat{Seg} = \arg \max_{Seg} P(Seg | Text)$$

$$= \arg \max_{Seg} \frac{P(Text | Seg) P(Seg)}{P(Text)}$$

?

$$\propto \arg \max_{Seg} P(Text | Seg) P(Seg)$$

$$= \arg \max_{Seg} P(Seg)$$

?

基于N元语法的切分排歧

- * Seg简写为S，内含n个词： $\{w_1, w_2, \dots, w_n\}$

$$p(S) = p(w_1^n) = p(w_1) \cdot \prod_{i=2}^n p(w_i | w_1^{i-1})$$

- * MM(马尔可夫模型/过程)：有限历史假设，第n个词的出现概率仅依赖前n-1个词
 - * 一种最简化的情况：一元文法/Uni-gram

$$P(S) = p(w_1) \cdot p(w_2) \cdot p(w_3) \dots p(w_n)$$

#连乘的代码实现?

基于N元语法的切分排歧

- * 采用一元语法：等价于最大频率分词
 - * 即把切分路径上每一个词的词频相乘得到该切分路径的概率
 - * 实际计算中：用词频的负对数，可以理解成一种“代价”
 - * 正确率可达到92%
 - * 简便易行，效果一般好于基于最大匹配的方法

n元语法(n-gram): 概念

- * 一元文法: $n=1$ 时, 出现在第 i 位上的词 ω_i 独立于历史, 记作unigram。
- * 二元文法: $n=2$ 时, 出现在第 i 位上的词 ω_i 只与前面的一个历史词 ω_{i-1} 有关, 记作bigram, 也被称为一阶马尔科夫链。
- * 三元文法: $n=3$ 时, 出现在第 i 位上的词 ω_i 只与与前面的两个历史词 $\omega_{i-1}\omega_{i-2}$ 有关, 记作trigram, 也被称作二阶马尔科夫链。

基于N元语法的切分排歧

- ❖ 采用二元语法：分词性能可以进一步提高

$$p(S) = p(w_1) \cdot p(w_2 | w_1) \cdot p(w_3 | w_2) \cdots p(w_n | w_{n-1})$$

- ❖ 采用更大的N：利用更多上下文信息

- ❖ 考虑参数空间

假设词表：20,000

n	n-gram的个数
2 (bigrams)	400,000,000
3 (trigrams)	8,000,000,000,000
4 (4-grams)	1.6×10^{17}

#解决参数爆炸的策略？得失？

基于N元文法的切分排歧

* 等价类映射：降低语言模型参数空间

- * 绝大多数历史不会出现在训练数据中。
- * 将历史 $\omega_1\omega_2...\omega_{i-1}$ 映射到等价类 $E(\omega_1\omega_2...\omega_{i-1})$ ，其中等价类的数目远小于全部历史数目。
- * 假设： $p(\omega_i|\omega_1...\omega_{i-1})=p(\omega_i|E(\omega_1\omega_2...\omega_{i-1}))$ ，则自由参数的数目会大大减少
- * 思考题：等价类的依据：必须符合语言学的分类

* 数据平滑（smoothing）：保持模型的辨别能力

- * 调整最大似然估计结果，更准确的估计未见事件
- * 提高低频率事件，降低高概率事件，概率分布更均匀
- * 课下专题阅读，统计自然语言处理，宗成庆，5.3节

目录

- * 语言模型的概念
- * 分词：从规则到统计的模型发展
- * n元语言模型
 - * n元文法
 - * 最大似然估计
 - * 语言模型性能评价
 - * 平滑
- * 神经网络语言模型
 - * 前馈神经网络语言模型
 - * 循环神经网络语言模型

n元语言模型

- 语言模型 (Language Model, LM)
 - 描述一段自然语言的概率或给定上文时下一个词出现的概率
 - $P(w_1, \dots, w_l), P(w_{l+1} | w_1, \dots, w_l)$
 - 以上两种定义等价 (链式法则)
 - $$P(w_1, \dots, w_l) = P(w_1)P(w_2 | w_1)P(w_3 | w_1 w_2) \dots P(w_l | w_1 w_2 \dots w_{l-1}) = \prod_{i=1}^l P(w_i | w_{1:i-1})$$
- 广泛应用于多种自然语言处理任务
 - 机器翻译 (词排序)
 - $P(\text{the cat is small}) > P(\text{small the is cat})$
 - 语音识别 (词选择)
 - $P(\text{there are four cats}) > P(\text{there are for cats})$

n元语言模型

- * 如前所述，对于n-gram模型中条件概率的估计可以使用基于频次的方法。以bigram模型为例，我们计算：

$$\begin{aligned} p(w_i|w_{i-1}) &= \frac{C(w_{i-1}w_i)}{\sum_w C(w_{i-1}w)} \\ &= \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \end{aligned}$$

- * 这种估计方法称为最大似然估计（Maximum Likelihood Estimation, MLE）

n元语言模型

- * 例6.1. 假设训练语料由以下三个句子构成：

⟨S⟩ 我 喜欢 读书 ⟨/S⟩

⟨S⟩ 她 不 喜欢 编程 ⟨/S⟩

⟨S⟩ 我 不 喜欢 吃 辣椒 ⟨/S⟩

- * 根据最大似然估计得到的部分bigram条件概率如下

$$p(\text{我} | \langle S \rangle) = \frac{2}{3} \quad p(\text{喜欢} | \text{不}) = \frac{2}{2} = 1 \quad p(\text{读书} | \text{喜欢}) = \frac{1}{3}$$

$$p(\text{她} | \langle S \rangle) = \frac{1}{3} \quad p(\langle /S \rangle | \text{读书}) = \frac{1}{2} \quad p(\text{吃} | \text{喜欢}) = \frac{1}{3}$$

n元语言模型

- * 根据公式（6.3），我们可以进一步计算句子的概率，例如：

$$\begin{aligned} & p(<S> \text{我 喜欢 读书} </S>) \\ &= p(\text{我} | <S>) p(\text{喜欢} | \text{我}) p(\text{读书} | \text{喜欢}) p(</S> | \text{读书}) \\ &= \frac{2}{3} \times \frac{1}{3} \times \frac{1}{3} \times 1 \\ &= \frac{2}{27} \end{aligned}$$

平滑

- * 基于最大似然估计的方法所得到的n-gram模型存在一个潜在而经常出现的缺点，即“零概率问题”。在例6.1中，如果我们根据给定语料计算句子“我吃辣椒”在bigram语言模型中的概率，由于bigram“我吃”未在训练语料中出现，所以：

$$p(\text{吃}|\text{我}) = \frac{C(\text{我 吃})}{C(\text{我})} = \frac{0}{1} = 0$$

- * 当测试句子含有未登录词（Out-of-Vocabulary, OOV），以及训练语料难以覆盖n-gram，等因素也会导致语言模型将会不合理地估计句子概率为0

平滑

- * 折扣法：从频繁出现的n-gram 中匀出一部分概率并分配给低频次（含零频次）的n-gram，从而使得整体概率分布趋于均匀
- * 加一（ δ ）平滑（也被称为拉普拉斯平滑）：假设所有n-gram的频次比实际出现的频次多1（ δ ）次。 δ 是为了防止对于低频次事件给出过高的概率估计，根据开发集进行调整

- * unigram:

$$p(w_i) = \frac{C(w_i) + 1}{N + |V|}$$

- * bigram:

$$p(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) + 1}{\sum_w (C(w_{i-1}w) + 1)} = \frac{C(w_{i-1}w_i) + 1}{C(w_{i-1}) + |V|}$$

平滑

- * 插值法：基本思想是当没有足够的语料估计高阶模型的概率时，低阶模型往往可以提供有用的信息
- * *Kneser-Ney* 平滑：是应用得较为广泛也是最有效的平滑算法之一，它结合了绝对折扣法与插值的思想，提供了一种新的融合高阶模型和低阶模型的方式
- * 这两种平滑方法请同学们课下进行阅读学习

语言模型性能评价

- * 如何评价语言模型的好坏？
- * 一种方法是将其应用于具体的外部任务（例如机器翻译），并且根据该任务上的指标的变化来对语言模型进行评价。这种方法也被称为外部任务评价
- * 但是外部任务评价的代价较高，所以我们现在最为常用的仍然是给予**困惑度**（Perplexity, PPL）的内部评价方式

语言模型性能评价

- * 为了进行内部评价，我们首先将数据分为不相交的两个集合分别称为训练集 D_{train} 和测试集 D_{test}
- * 其中 D_{train} 用于估计语言模型的参数。由该模型计算出的测试集的概率 $p(D_{test})$ 则反映了模型在测试集上的泛化能力
- * 当模型较为复杂（例如使用了平滑技术）时，在测试集上反复评价并调整超参数的方式会使得模型在一定程度上拟合了测试集。因此在标准实践中，需要划分一个额外的集合以用于训练过程中的必要调试。该集合通常称为开发集（development set），也称验证集（validation set）

语言模型性能评价

- * 定义6.1. 假设数据集为 $D_{test} = w_1 w_2 \dots w_N$ （句子间由<S>和</S>分割），那么测试集的概率为：

$$\begin{aligned} p(\mathcal{D}_{test}) &= p(w_1 w_2 \dots w_N) \\ &= \prod_{i=1}^N p(w_i | w_1^{i-1}) \end{aligned}$$

N为测试集总词数，而不是某一个句子数，由于我们一旦对训练集测试集都全连上一定会估计出 $p(<S> | </S>) = 1$ ，所以可以直接连接，和断开相乘无区别

- * 困惑度则为模型分配给测试集中每一个词的几何平均值的代数：

$$\text{PPL}(\mathcal{D}_{test}) = \left(\prod_{i=1}^N p(w_i | w_1^{i-1}) \right)^{-\frac{1}{N}}$$

语言模型性能评价

- * 对于bigram模型而言，式子变为：

$$\text{PPL}(\mathcal{D}_{test}) = \left(\prod_{i=1}^N p(w_i | w_{i-1}) \right)^{-\frac{1}{N}}$$

- * 在实际计算中，考虑到多个概率的连乘可能带来的浮点数下溢的问题，通常需要将上式转化为对数和的形式：

$$\text{PPL}(\mathcal{D}_{test}) = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2 p(w_i | w_{i-1})}$$

测试集中已然都是“人话”，困惑度越小意味着单词序列的概率越大，也意味着模型能够更好地解释测试集中的数据。

目录

- * 语言模型的概念
- * 分词：从规则到统计的模型发展
- * n元语言模型
 - * n元文法
 - * 最大似然估计
 - * 语言模型性能评价
 - * 平滑
- * 神经网络语言模型
 - * 前馈神经网络语言模型
 - * 循环神经网络语言模型

神经网络语言模型

- * **n-gram语言模型的明显缺点：**
- * 高阶n元语言模型面临着严重的数据稀疏问题，虽然可以使用平滑来进行近似，但是也导致了长历史信息无法被充分地利用
- * n元模型忽略了词与词之间的相似性，导致泛化能力不足。如 “The cat is walking in the bedroom” 和 “A dog was running in a room”，因为对应位置在句中充当的句法和语义角色都非常相近，显然模型应该给出类似的概率估计
- * 神经网络语言模型，通过使用分布式表示在一定程度上克服了上述问题

前馈神经网络语言模型

- * 在前馈神经网络语言模型中，我们仍然沿用n元语言模型的假设。我们的目标仍然是估计条件概率 $p(w_t | w_{t-n+1}^{t-1})$ 。如果将词表 V 中的每个词看成一个类别，那么对该条件概率的估计可以看作 V 上的一个分类问题，从而可以使用机器学习中用于分类的概率模型进行解决
- * 前馈神经网络语言模型（Feed-forward Neural Network Language Model），结合词的分布式表示以及前馈神经网络实现了对于条件概率 $p(w_t | w_{t-n+1}^{t-1})$ 的估计

前馈神经网络语言模型

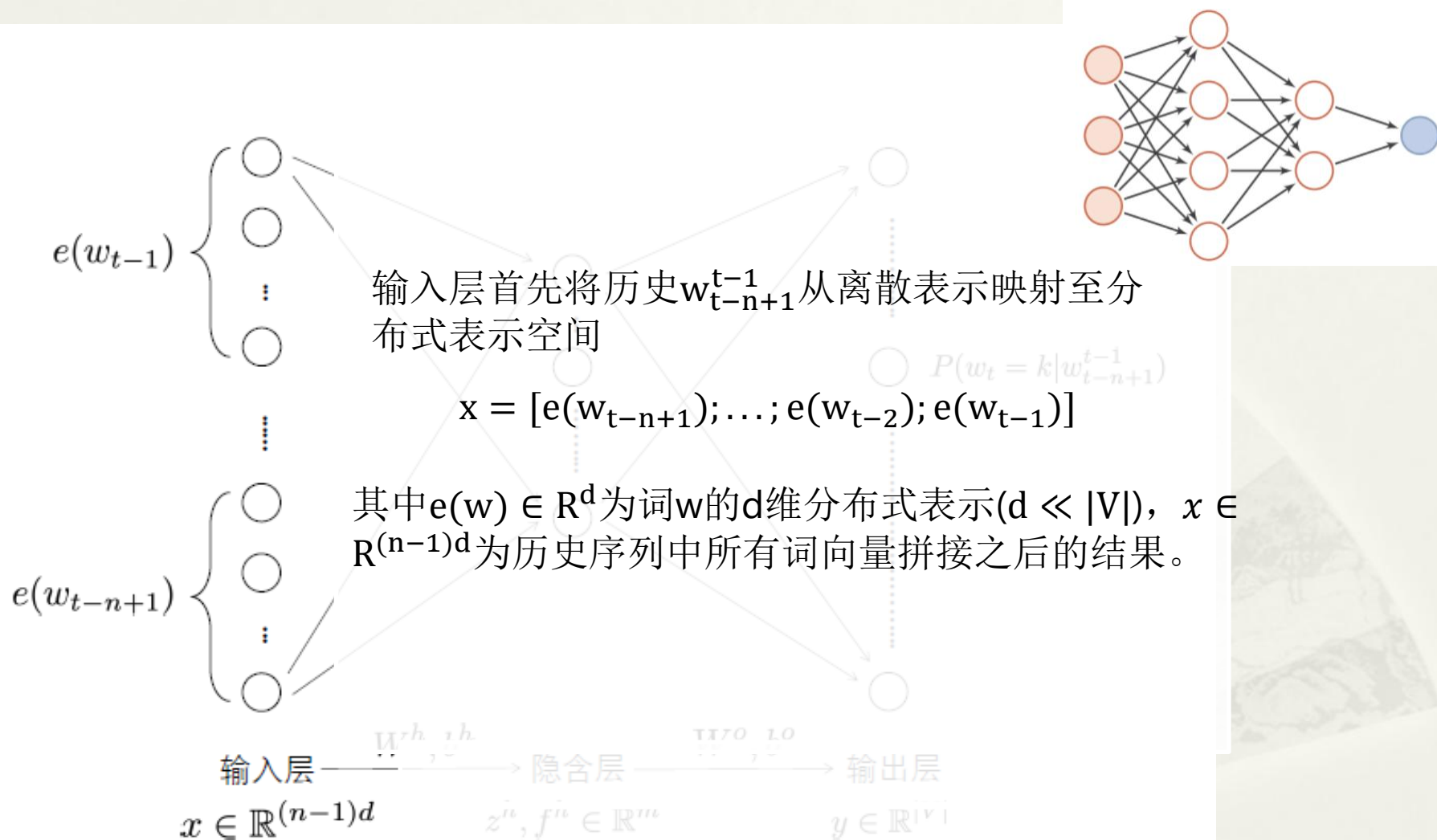


图 6.1 前馈神经网络语言模型结构。

前馈神经网络语言模型

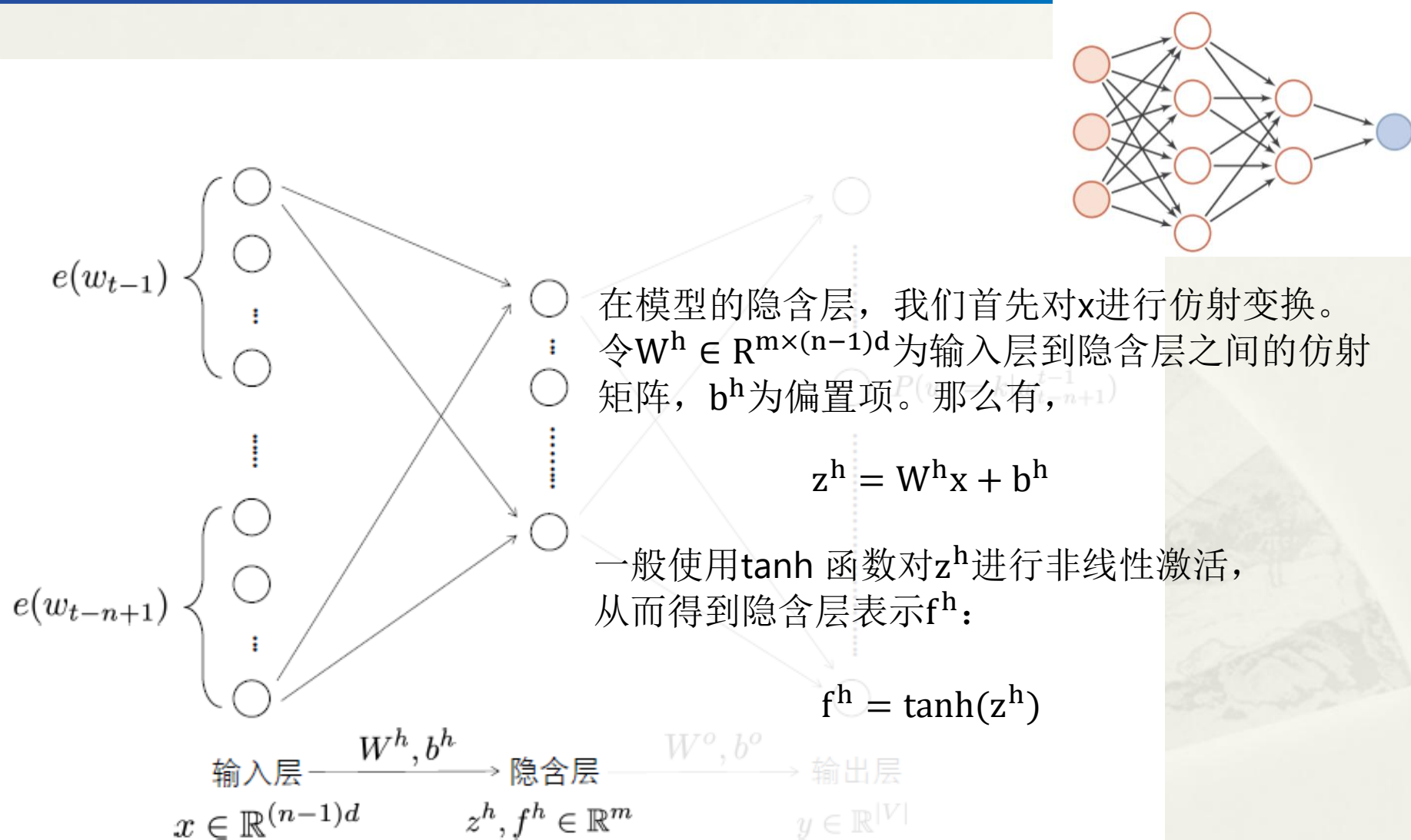
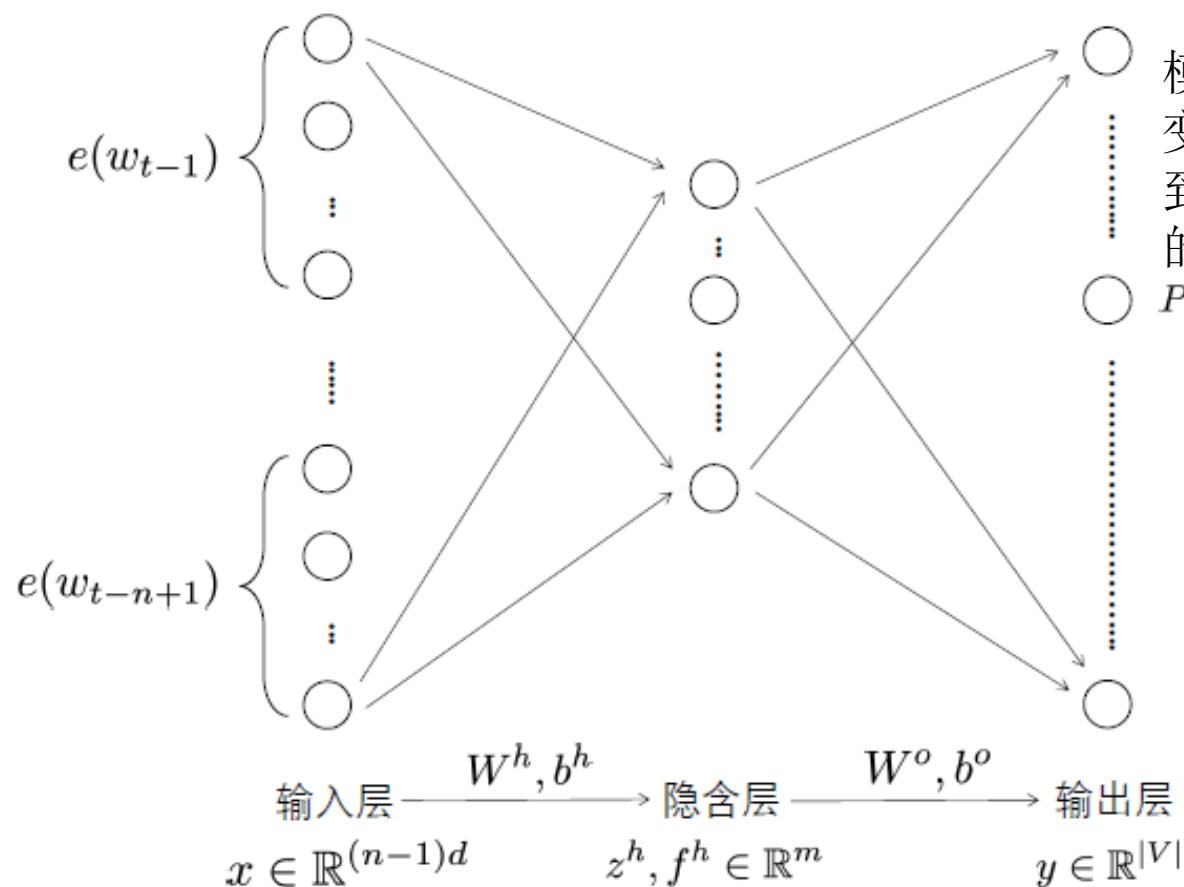


图 6.1 前馈神经网络语言模型结构。

前馈神经网络语言模型



模型的输出层再次对 f^h 进行仿射变换。令 $W^o \in \mathbb{R}^{|V| \times m}$ 是隐含层到输出层之间的仿射矩阵，相应的偏置项为 b^o ，那么，

$$P(w_t = k | w_{t-n+1}^{t-1})$$

$$z^o = W^o f^h + b^o$$

利用softmax函数对 z^o 进行归一化，从而获得在词表 V 上的概率分布

$$y = \text{softmax}(z^o)$$

下一个词 w_t 是词表中第 k 个词 ($k \in \{1, \dots, |V|\}$) 的概率

$$p(w_t = k | w_{t-n+1}^{t-1}) = y_k = \frac{e^{z_k^o}}{\sum_{l=1}^{|V|} e^{z_l^o}}$$

图 6.1 前馈神经网络语言模型结构。

循环神经网络语言模型

- * 前馈神经网络语言模型中对下一个词的预测需要回看历史的长度是由超参 n 来决定的。其实不然
 - * “He eats an apple”，预测apple要向前看到eat
 - * “I saw the ship with very strong binoculars”，预测binoculars需要看到saw
- * 循环神经网络语言模型（Recurrent Neural Network Language Model）正是为了处理这种不定长依赖而设计的一种语言模型。
 - * 用来处理时序数据的一种神经网络，而自然语言正好满足这种序列性质。
 - * 循环神经网络语言模型中的每一时刻都维护一个隐含状态，该状态蕴含了当前词的所有历史信息，且与当前词一起被作为下一时刻的输入

循环神经网络语言模型

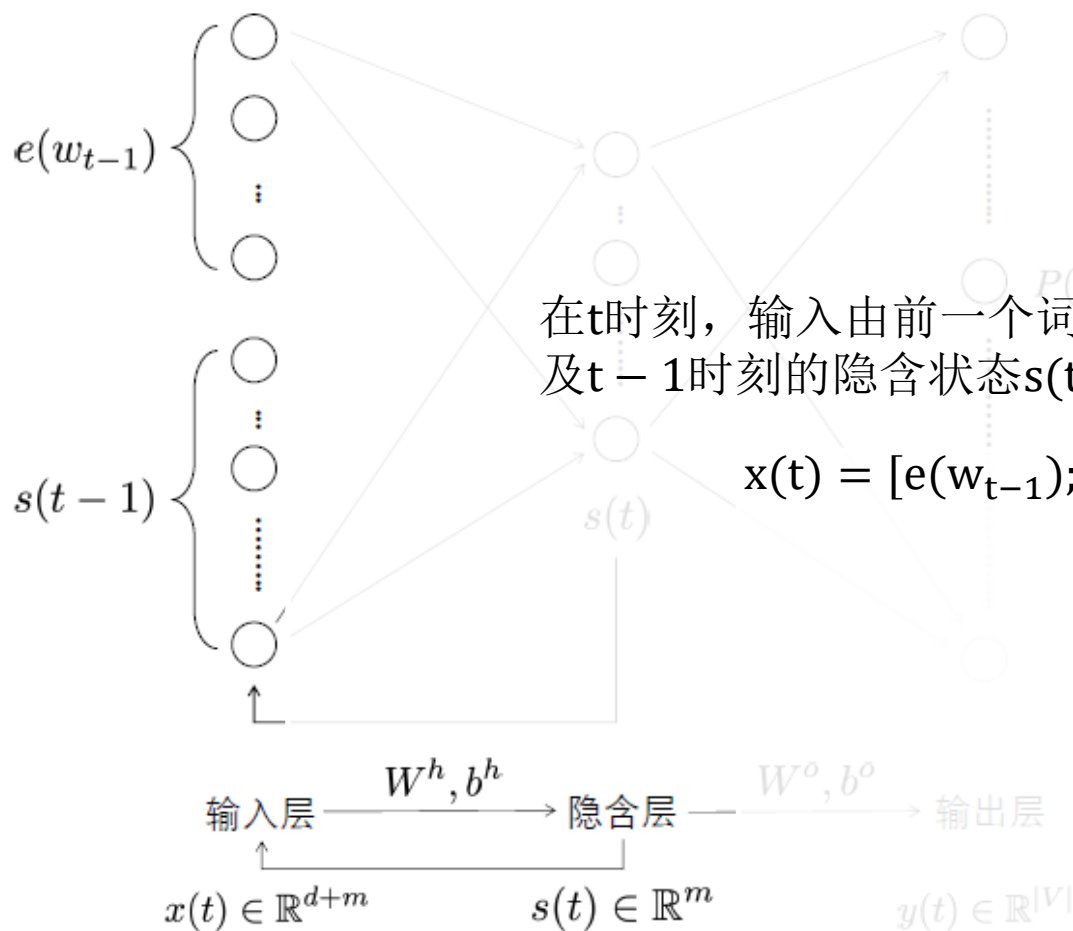
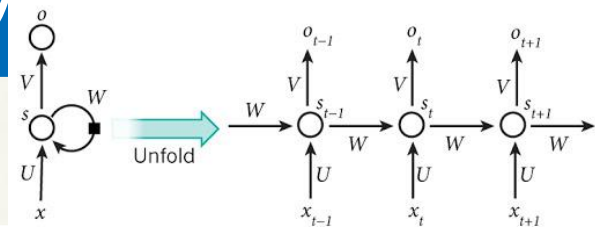


图 6.2 循环神经网络语言模型。

循环神经网络语言模型

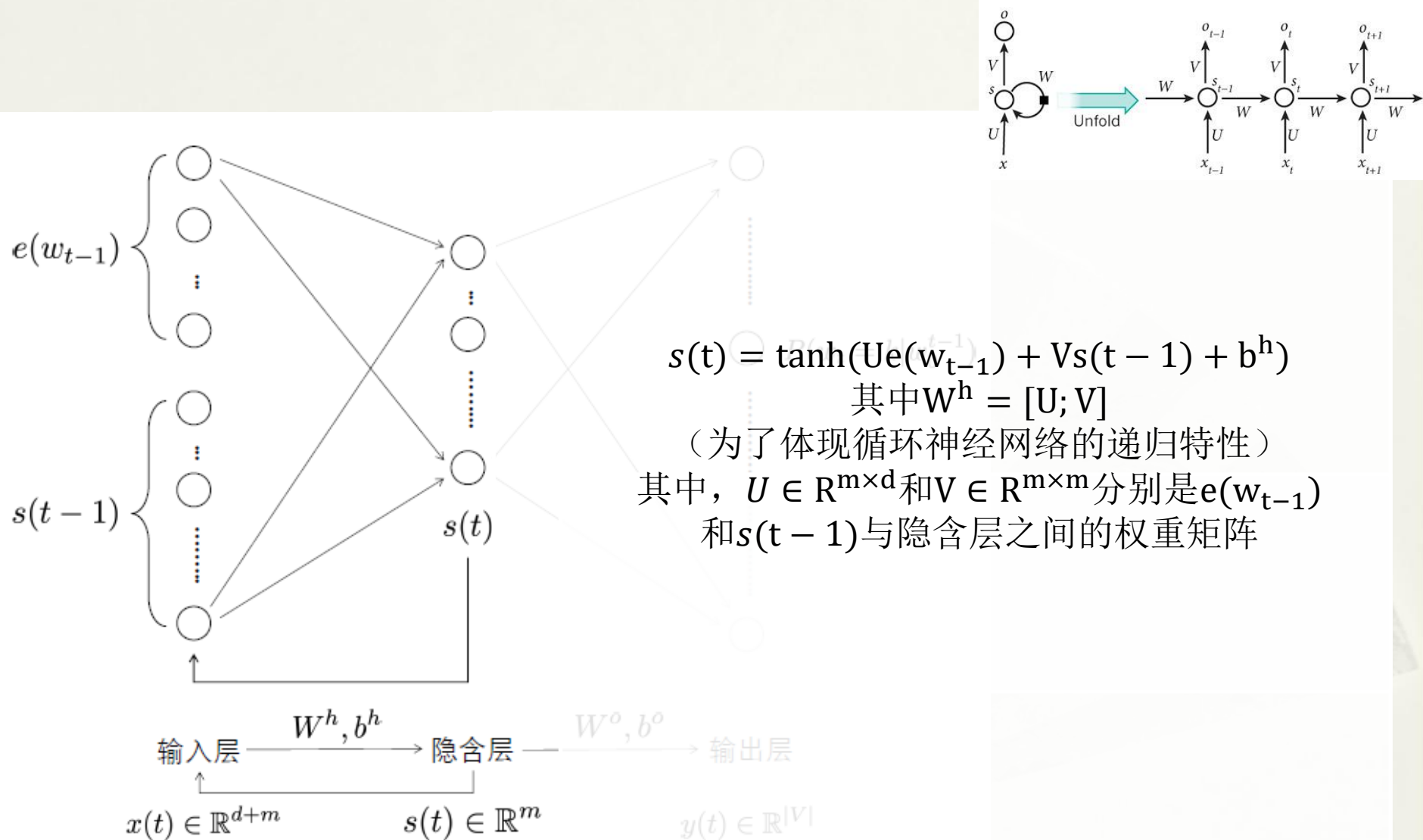


图 6.2 循环神经网络语言模型。

循环神经网络语言模型

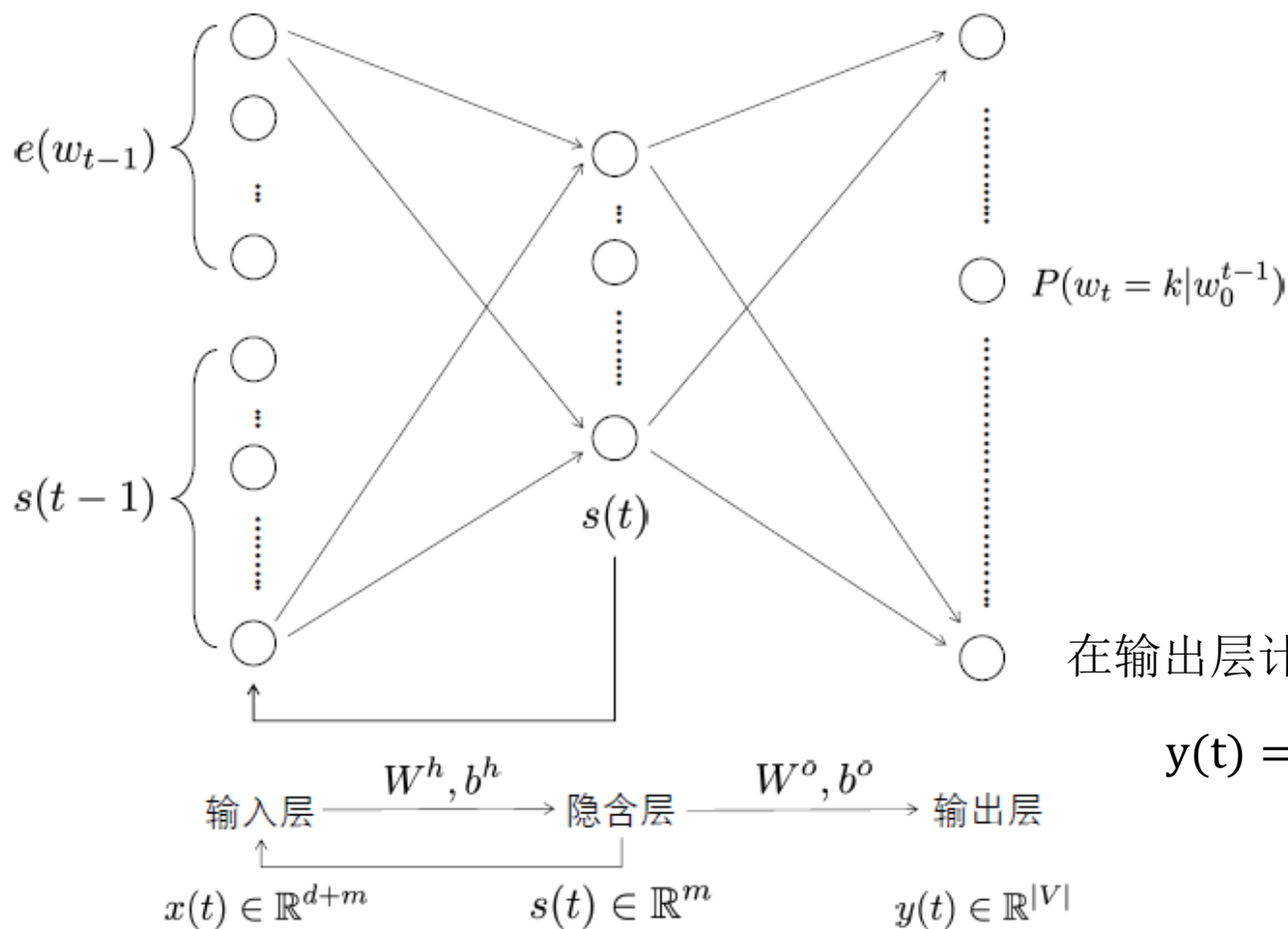
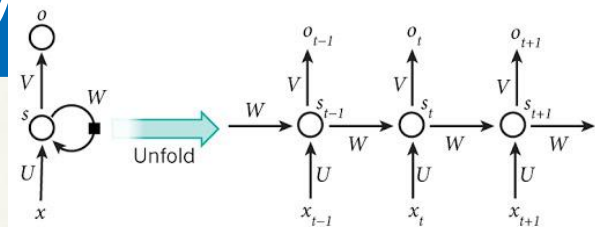
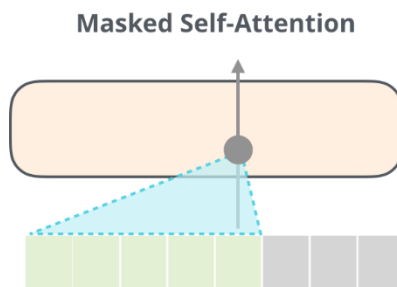
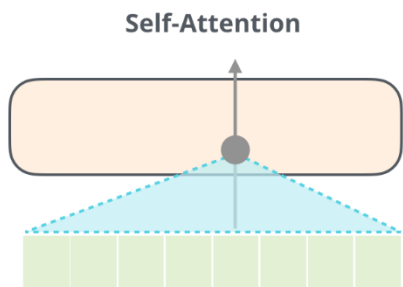


图 6.2 循环神经网络语言模型。

▣ 使用Transformer估计条件概率

$$P(w_t|w_{1:t-1}) \approx P(w_t|w_{t-n+1:t-1})$$



Features				Labels
position:	1	2	3	4
Example:				
1	robot	must	obey	orders
2	robot	must	obey	orders
3	robot	must	obey	orders
4	robot	must	obey	orders

must
obey
orders
<eos>

参数估计

- * 前馈神经网络语言模型和循环神经网络语言模型的参数都可以通过最大似然法来估计。
 - * （最大化对数似然（机器学习角度）和最小化损失函数（深度学习角度）是等价的）

$$\begin{aligned}\mathcal{L}(\theta|s) &= f_{\theta}(s) = \log \prod_{i=1}^{n_s} p(w_i | w_{i-n+1}^{i-1}) \\ &= \sum_{i=1}^{n_s} \log p(w_i | w_{i-n+1}^{i-1}),\end{aligned}$$

- * 对于目标函数的优化可以采用随机梯度下降法进行优化，结合反向传播算法，可以有效地计算出 θ 中每个参数的梯度值，反复迭代，直到目标函数值不再下降或者达到预设的最大迭代次数为止

$$\theta = \theta - \eta \frac{\partial(\log p(w_i | w_{i-n+1}^{i-1}))}{\partial \theta},$$

总结

- * 语言模型（也称统计语言模型），是用来描述自然语言概率分布的模型
- * 给定一个词序列，语言模型根据习得的统计规律给出这个词序列作为一句话被产生的概率
- * 利用语言模型，也可以在给定上文条件下对接接下来可能出现的词进行预测
- * 同时语言模型还为自然语言的表示学习提供了天然的自监督优化目标