词性标注与语言模型

李辰

cli@xjtu.edu.cn

2024年9月

2/3

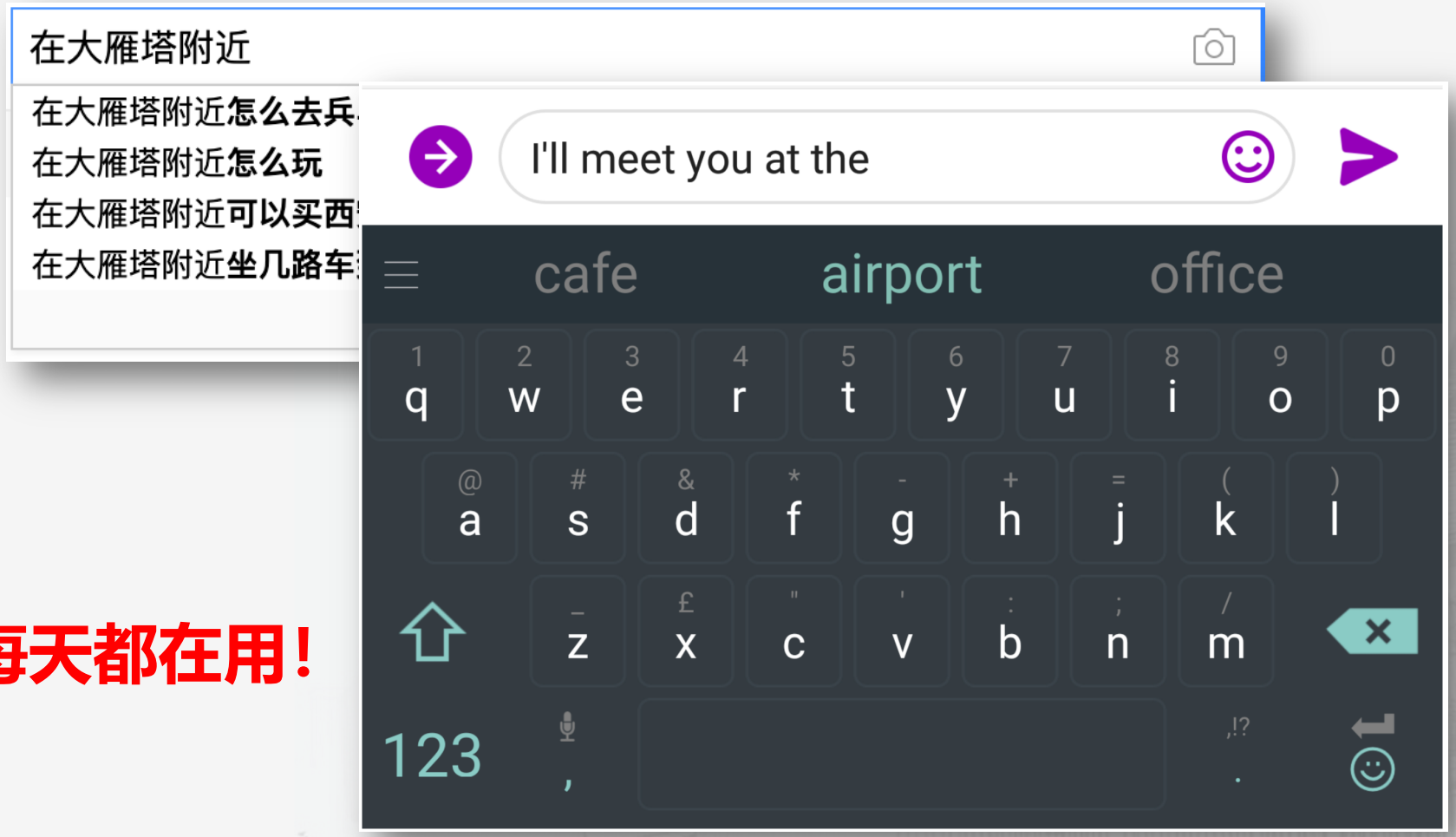
课程提纲

一、词性标注

二、语言模型

语言模型

- 回忆一下，你是否用过语言模型?



每天都在用!

本讲概览

一个新的自然语言处理任务

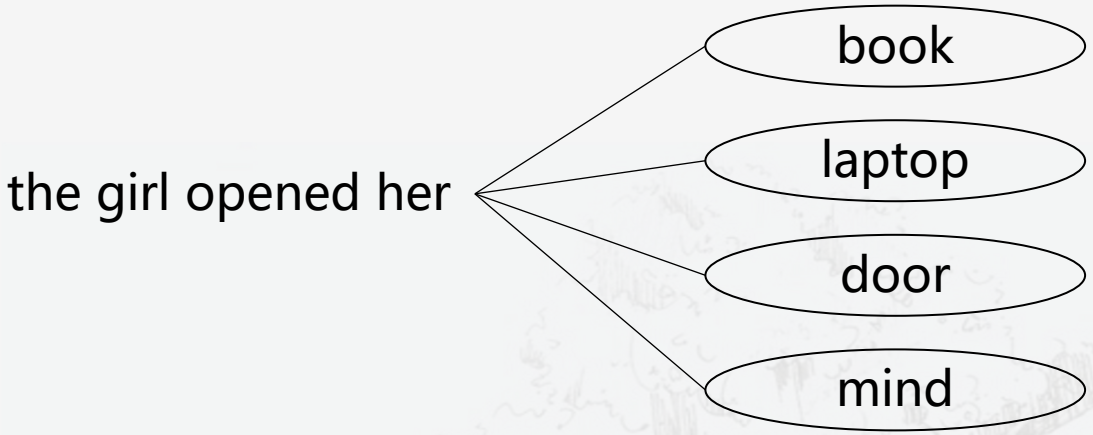
- 语言模型（language modeling）

相关概念

- 马尔可夫链（Markov chain）
- n-gram模型
- 最大似然估计（Maximum Likelihood Estimation, MLE）
- 拉普拉斯平滑处理（Laplace smoothing, additive smoothing）
- Good Turing

语言模型

- 定义：语言模型是用来计算一个句子的概率的概率模型
- 用途：已知一个句子已有的词，预测下一个词。



- 正式定义：

  给定一个词序列 $x^{(1)}, x^{(2)}, \ldots, x^{(t)}$，计算下一个可能出现的词的概率分布 $x^{(t+1)}$。

$$P\left(x^{(t+1)} | x^{(t)}, \ldots, x^{(1)}\right)$$

  其中 $x^{(t+1)}$ 是词汇表 $V = \{w_1, \ldots, w_{|V|}\}$ 中的任意一个词。

- 这样的一个系统就是一个语言模型。

语言模型

- 语言模型也可以被理解为一个对一段文本指定概率的系统。
- 例如：文本 $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ 的概率（语言模型）可以记为

$$\mathrm{P}\left(x^{(1)}, \ldots, x^{(T)}\right) = P\left(x^{(1)}\right) \times P\left(x^{(2)} \middle| x^{(1)}\right) \times \cdots \times P\left(x^{(T)} \middle| x^{(T-1)}, \ldots, x^{(1)}\right)$$

$$= \prod_{t=1}^{T} P\left(x^{(t)} \middle| x^{(t-1)}, \ldots, x^{(1)}\right)$$

这是语言模型所能提供的

n-gram 语言模型

the girl opened her _____

- 问题：如何训练一个语言模型?
- 理想情况下：$\prod_{t=1}^{T} P\left(x^{(t)} \middle| x^{(t-1)}, \ldots, x^{(1)}\right)$ 计算量巨大！！！
- n-gram 语言模型定义：一个 n-gram 是一个包含 n个连续词的文本。
  - unigrams:
  - bigrams:
  - trigrams:
  - 4-grams:
  - ...
- 思路：先收集每一组 n-gram词的出现频率，然后用这些频率预测将要出现的下一个词。

n-gram 语言模型

the girl opened her _____

- 首先做一个简化的假设：$x^{(t-1)}$ 的预测只依赖于之前的 $n-1$ 个词

$$P\left(x^{(t+1)}\middle|x^{(t)},\dots,x^{(1)}\right)=P\left(x^{(t+1)}\middle|\underbrace{x^{(t)},\dots,x^{(t-n+2)}}_{n-1\text{ 个词}}\right)$$

假设

n-gram 概率

$(n-1)$-gram 概率

$$=\frac{P\left(x^{(t+1)},x^{(t)},\dots,x^{(t-n+2)}\right)}{P\left(x^{(t)},\dots,x^{(t-n+2)}\right)}$$

条件概率

- 问题：如何计算 n-gram 和 $(n-1)$-gram 的概率?
- 方法：在一个大的语料库（corpus）上计算出现的次数。

$$\approx\frac{count\left(x^{(t+1)},x^{(t)},\dots,x^{(t-n+2)}\right)}{count\left(x^{(t)},\dots,x^{(t-n+2)}\right)}$$

统计近似值

n-gram 语言模型

- 假设我们训练一个 4-gram 的语言模型

~~*Before her mum arrives, the*~~ *girl opened her* _____

条件

$$P(\boldsymbol{w}|girl\ opened\ her) = \frac{count(girl\ opened\ her\ \boldsymbol{w})}{count(girl\ opened\ her)}$$

例如：在语料库里

- "*girl opened her*" 出现了1000次
- "*girl opened her book*" 出现了400次

$$P(\boldsymbol{book}|girl\ opened\ her) = 0.4$$

- "*girl opened her laptop*" 出现了100次

$$P(\boldsymbol{laptop}|girl\ opened\ her) = 0.1$$

应该完全丢弃上下文"*Before her mum arrives*"吗?

n-gram 模型稀疏问题

问题1：如果数据中不包含“*girl opened her* ***w***”

解决方案：给每一个词 ***w*** 的计数加一。这种方法被称为平滑处理（smoothing）

$$P(\boldsymbol{w}|girl\ opened\ her) = \frac{count(girl\ opened\ her\ \boldsymbol{w})}{count(girl\ opened\ her)}$$

问题2：如果数据中不包含“*girl opened her*”

解决方案：条件只设为“*opened her*”。这种方法称为回退（backoff）。

思考：增大 n，能否缓解稀疏问题？ n一般不会大于 5。

n-gram 存储问题

问题：需要存储所有的 n-gram

$$P(\boldsymbol{w}|girl\ opened\ her) = \frac{count(girl\ opened\ her\ \boldsymbol{w})}{count(girl\ opened\ her)}$$

思考：增大 n 或增大语料集，会导致模型规模增大。

基于n-gram的文本生成

today the _____

基于n-gram的文本生成

例子

- 构建一个 trigram 语言模型；
- 170万词路透社语料库（corpus）

商业与金融新闻

today the _____

Ref: https://nlpforhackers.io/language-models/

基于n-gram的文本生成

例子

- 构建一个 trigram 语言模型；
- 170万词路透社语料库（corpus）

商业与金融新闻

today the _____

计算分布概率

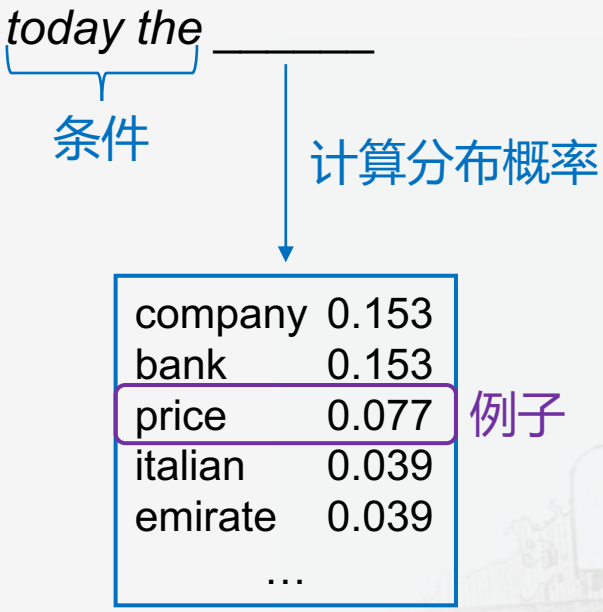
company	0.153
bank	0.153
price	0.077
italian	0.039
emirate	0.039
…	

稀疏问题

基于n-gram的文本生成

例子

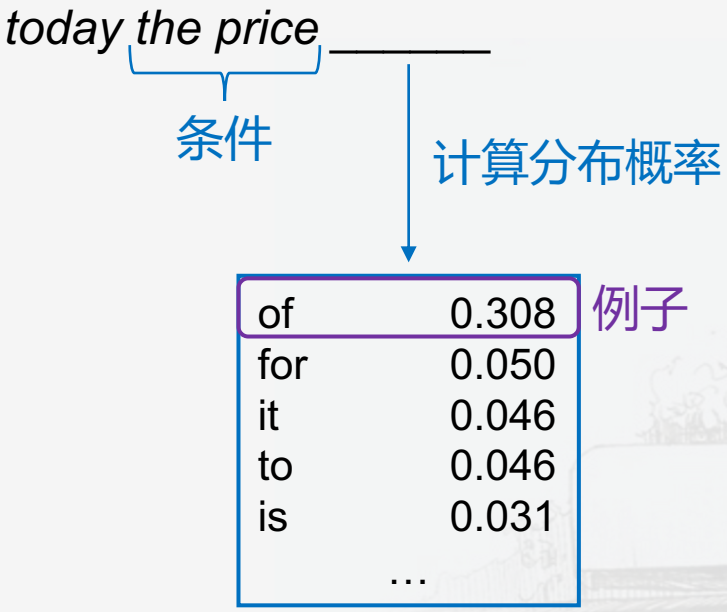
- 使用语言模型生成文本。



基于n-gram的文本生成

例子

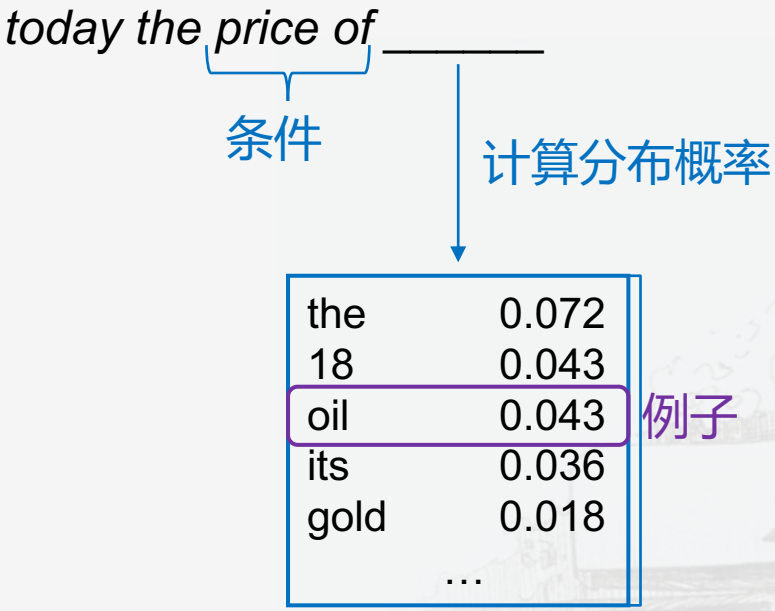
- 使用语言模型生成文本。



基于n-gram的文本生成

例子

- 使用语言模型生成文本。



基于n-gram的文本生成

例子

- 使用语言模型生成文本。

today the price of oil

基于n-gram的文本生成

例子

- 使用语言模型生成文本。

today the price of oil per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

- 语法正确
- 内容不连贯

基于n-gram的文本生成

例子

- 使用语言模型生成文本。

today the price of oil per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

- 语法正确
- 内容不连贯　　→ 模型需要考虑超过3个词，从而获得更连贯的内容。

基于n-gram的文本生成

例子

- 使用语言模型生成文本。

today the price of oil per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

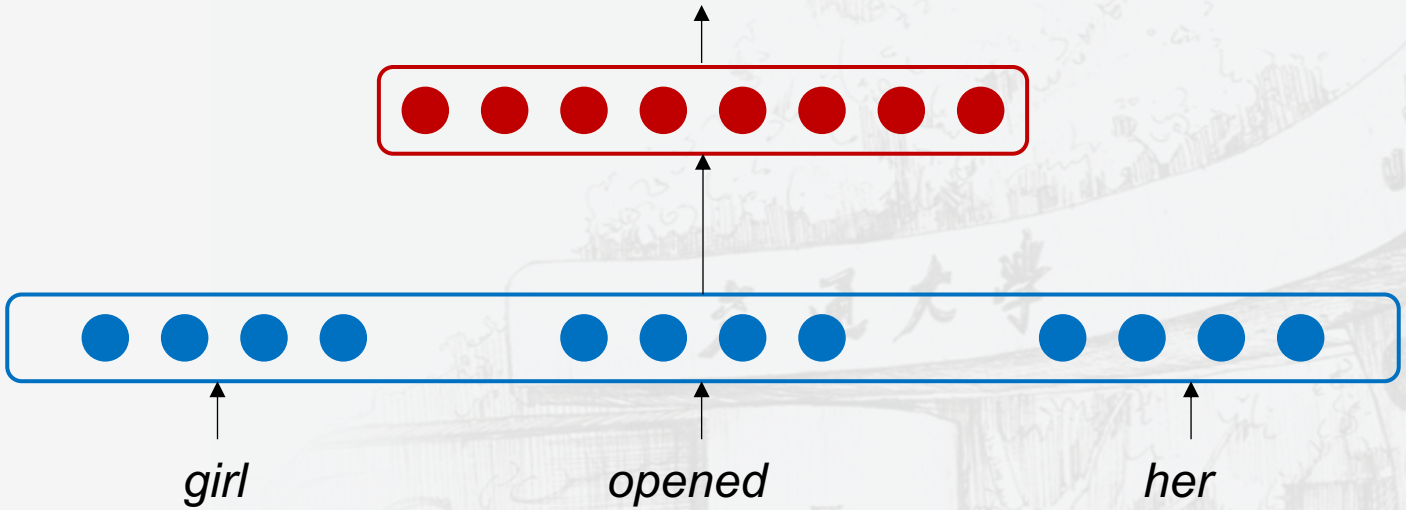
- 语法正确
- 内容不连贯 → 模型需要考虑超过3个词，从而获得更连贯的内容。
  → 增加 n 加剧稀疏问题，同时增大模型体积。

基于神经网络的语言模型

语言模型任务回顾

- 输入：词序列 $x^{(1)}, x^{(2)}, \ldots, x^{(t)}$
- 输出：下一个词的概率分布 $P(x^{(t+1)}|x^{(t)}, \ldots, x^{(1)})$

基于窗口的神经网络模型?



基于神经网络的语言模型

~~*Before her mum arrives*~~ *the girl opened her* _____

舍弃

固定窗口

基于神经网络的语言模型

词向量（one-hot、分布式表示 ……）

$x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$

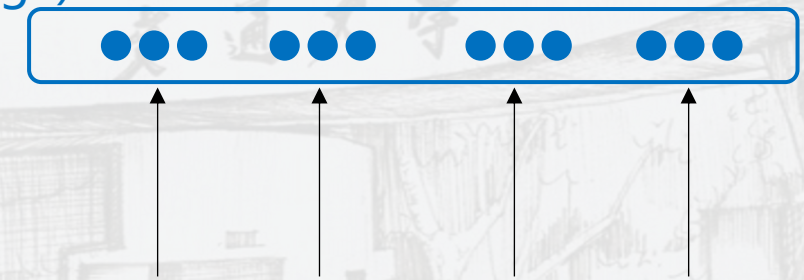
the *girl* *opened* *her* _____

$x^{(1)}$ $x^{(2)}$ $x^{(3)}$ $x^{(4)}$

基于神经网络的语言模型

连接嵌入式词向量（word embeddings）

$$e = [e^{(1)};\ e^{(2)};e^{(3)};\ e^{(4)}]$$



词向量（one-hot、分布式表示 ……）

$$x^{(1)}, x^{(2)}, x^{(3)},\ x^{(4)}$$

基于神经网络的语言模型

隐藏层

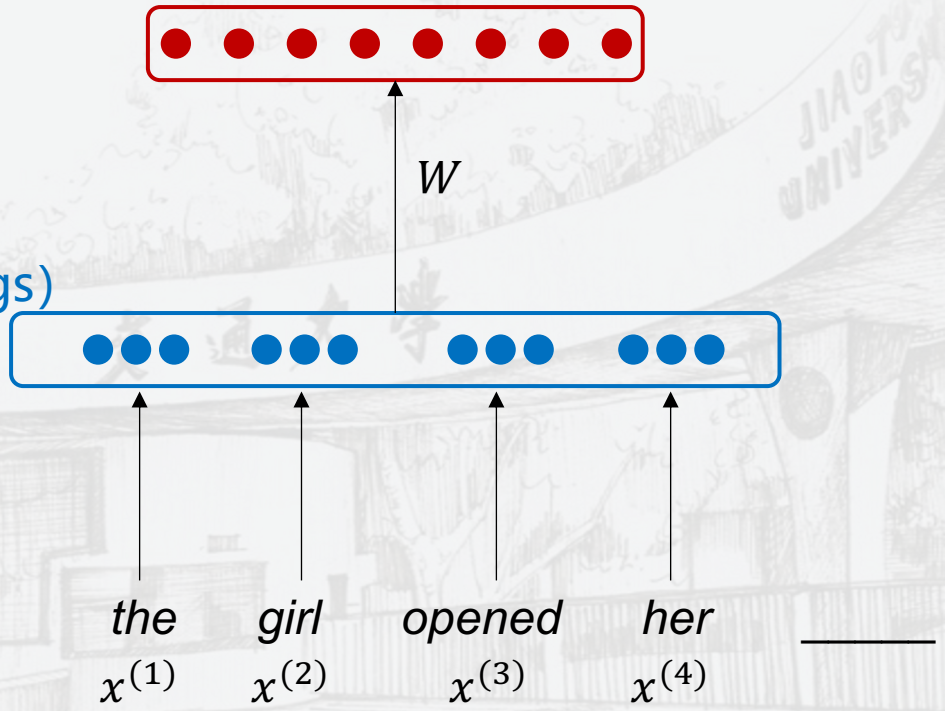
$$h = f(We + b_1)$$

连接嵌入式词向量（word embeddings）

$$e = [e^{(1)}; e^{(2)}; e^{(3)}; e^{(4)}]$$

词向量（one-hot、分布式表示 ……）

$$x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$$



基于神经网络的语言模型

输出层

$$\hat{y} = softmax(Uh + b_2) \in \mathbb{R}^{|V|}$$

隐藏层

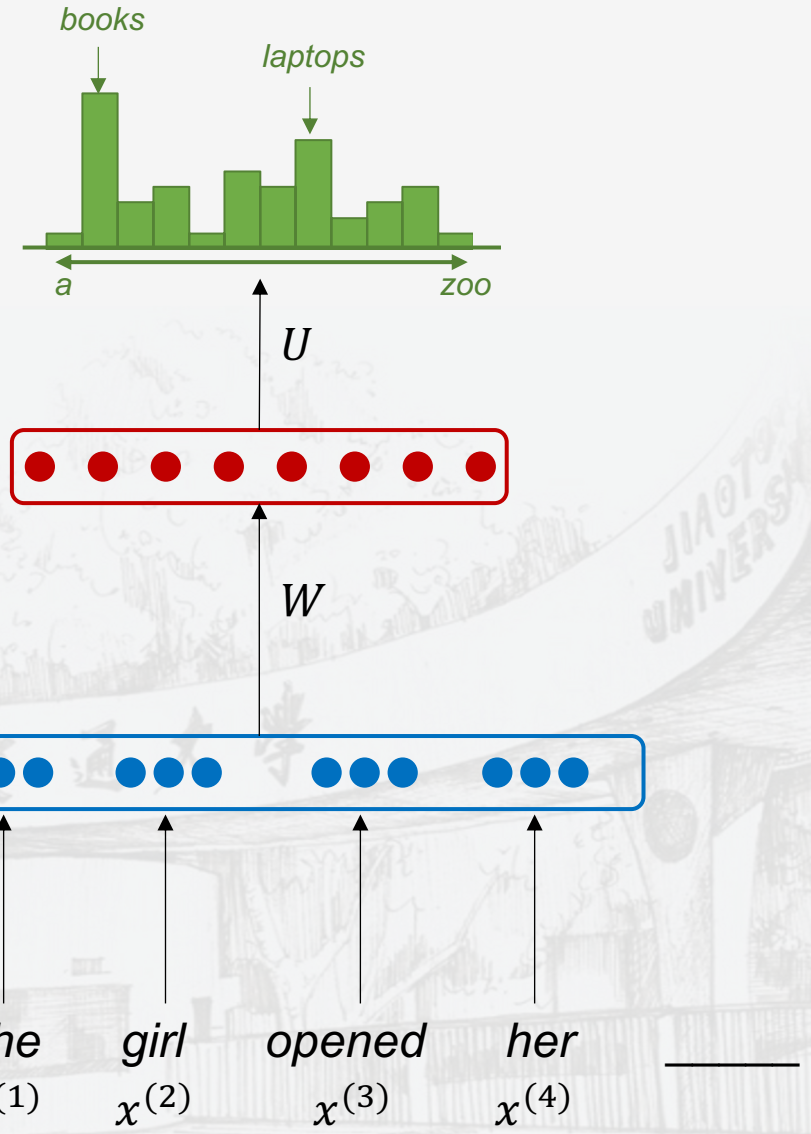
$$h = f(We + b_1)$$

连接嵌入式词向量（word embeddings）

$$e = [e^{(1)}; e^{(2)}; e^{(3)}; e^{(4)}]$$

词向量（one-hot、分布式表示 ……）

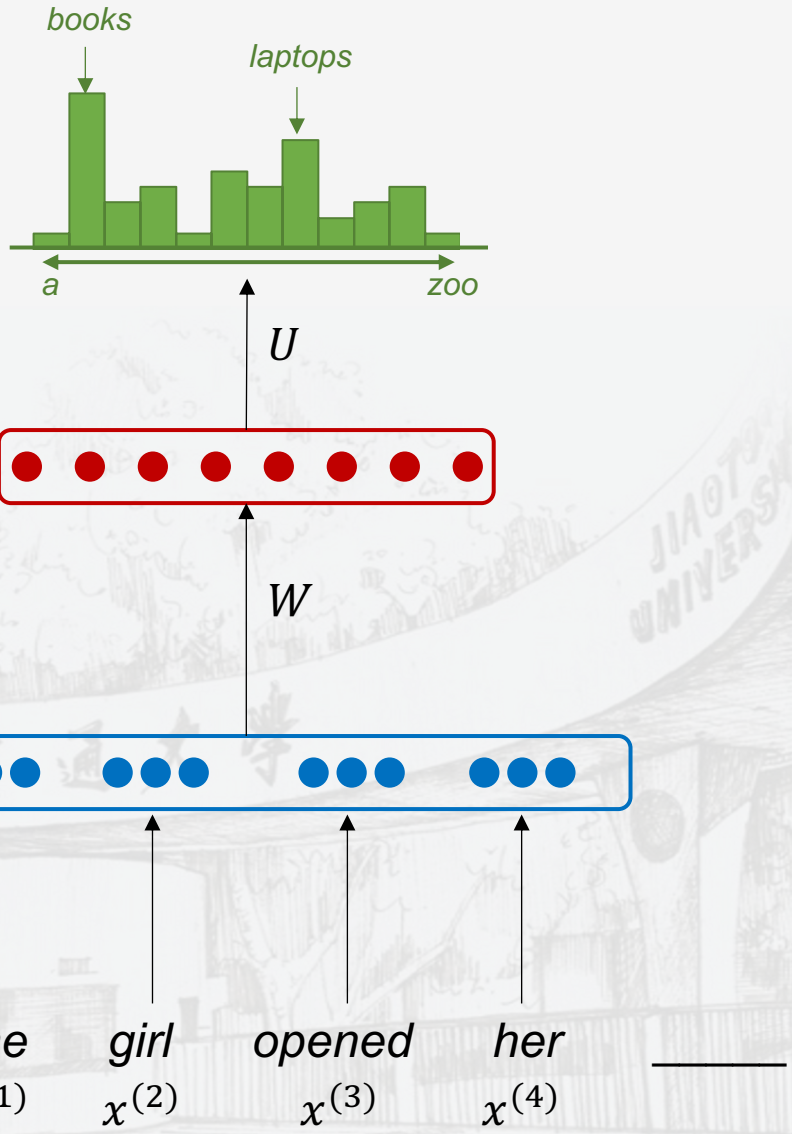
$$x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$$



基于神经网络的语言模型

与 n-gram 模型相比的优势

- 没有稀疏问题
- 无需存储所有观察到的 n-gram



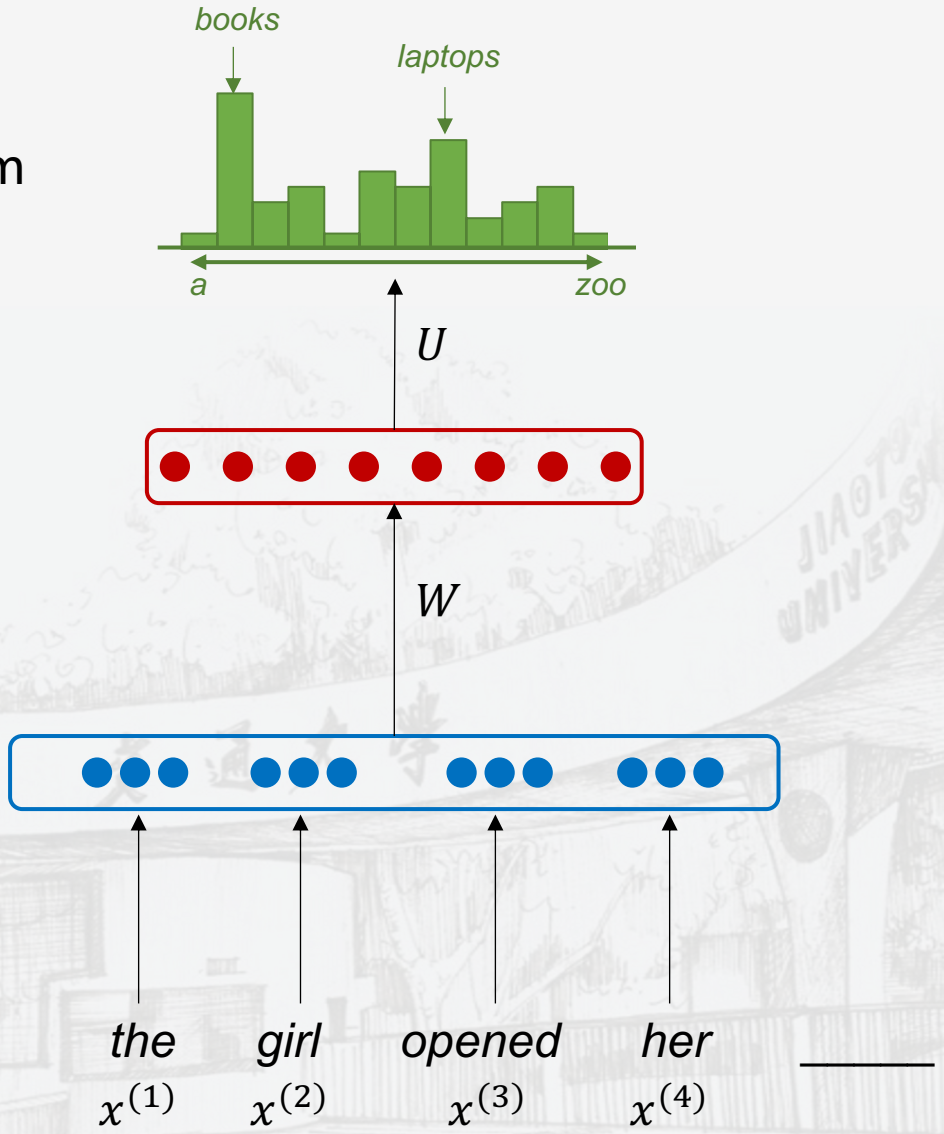
基于神经网络的语言模型

与 n-gram 模型相比的优势

- 没有稀疏问题
- 无需存储所有观察到的 n-gram

问题

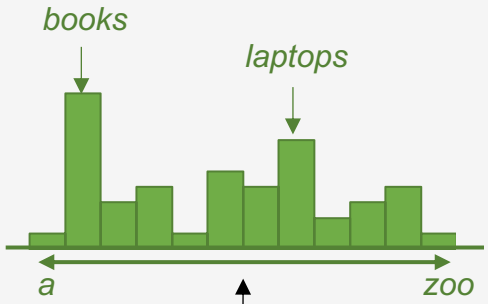
- 窗口太小
- 窗口值扩大导致 W 增大
- 窗口永远无法足够大!



基于神经网络的语言模型

与 n-gram 模型相比的优势

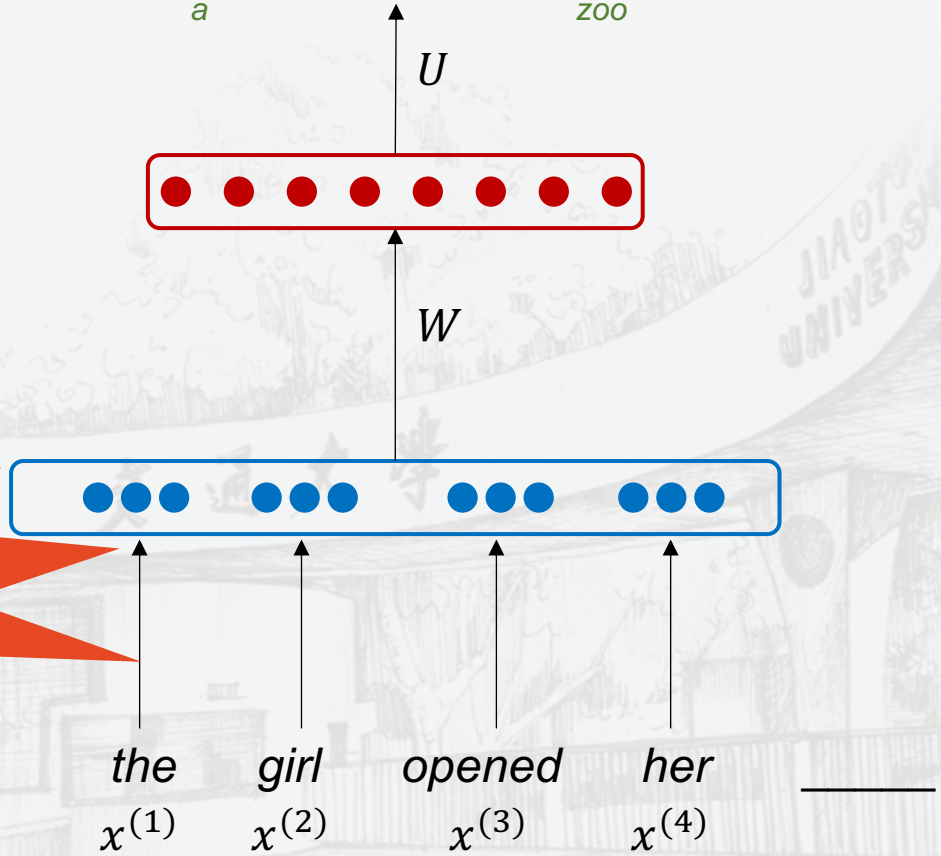
- 没有稀疏问题
- 无需存储所有观察到的 n-gram



问题

- 窗口太小
- 窗口值扩大导致 W 增大
- 窗口永远无法足够大!

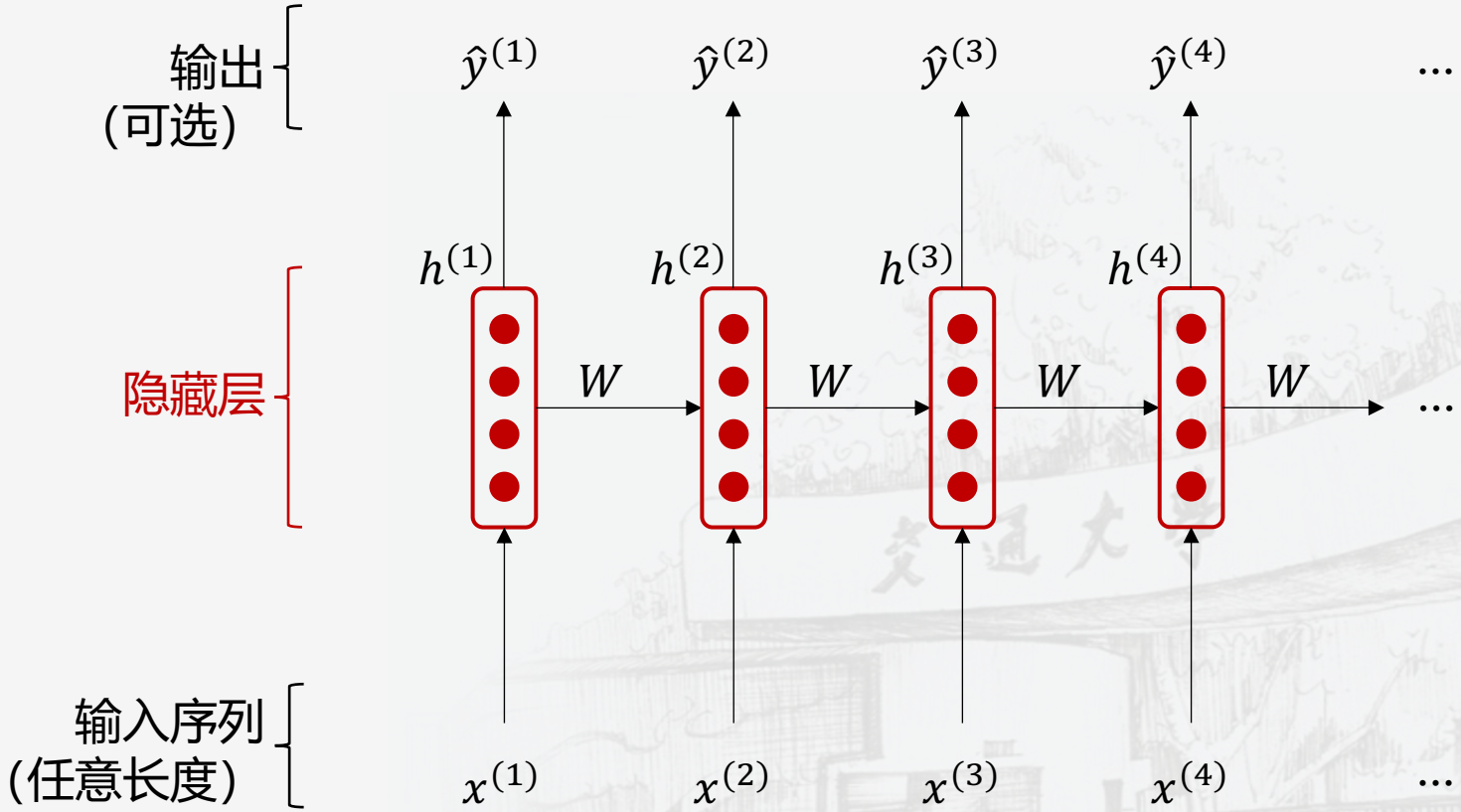
需要一个能够处理任意长度的神经网络结构



基于递归神经网络的LM

Recurrent Neural Networks (RNN)

- 核心思想：使用同一个 W



基于递归神经网络的LM

词向量（one-hot、分布式表示 .....）

$x^{(t)} \in \mathbb{R}^{|V|}$

the	*girl*	*opened*	*her*
$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$

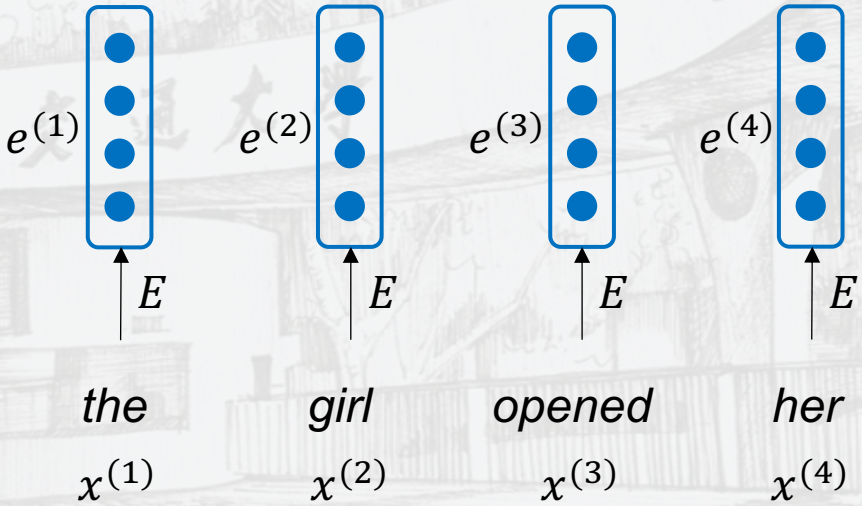
基于递归神经网络的LM

嵌入式词向量

$e^{(t)} = Ex^{(t)}$

词向量（one-hot、分布式表示 .....）

$x^{(t)} \in \mathbb{R}^{|V|}$



基于递归神经网络的LM

隐藏层

$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e^{(t)} + b_1)$

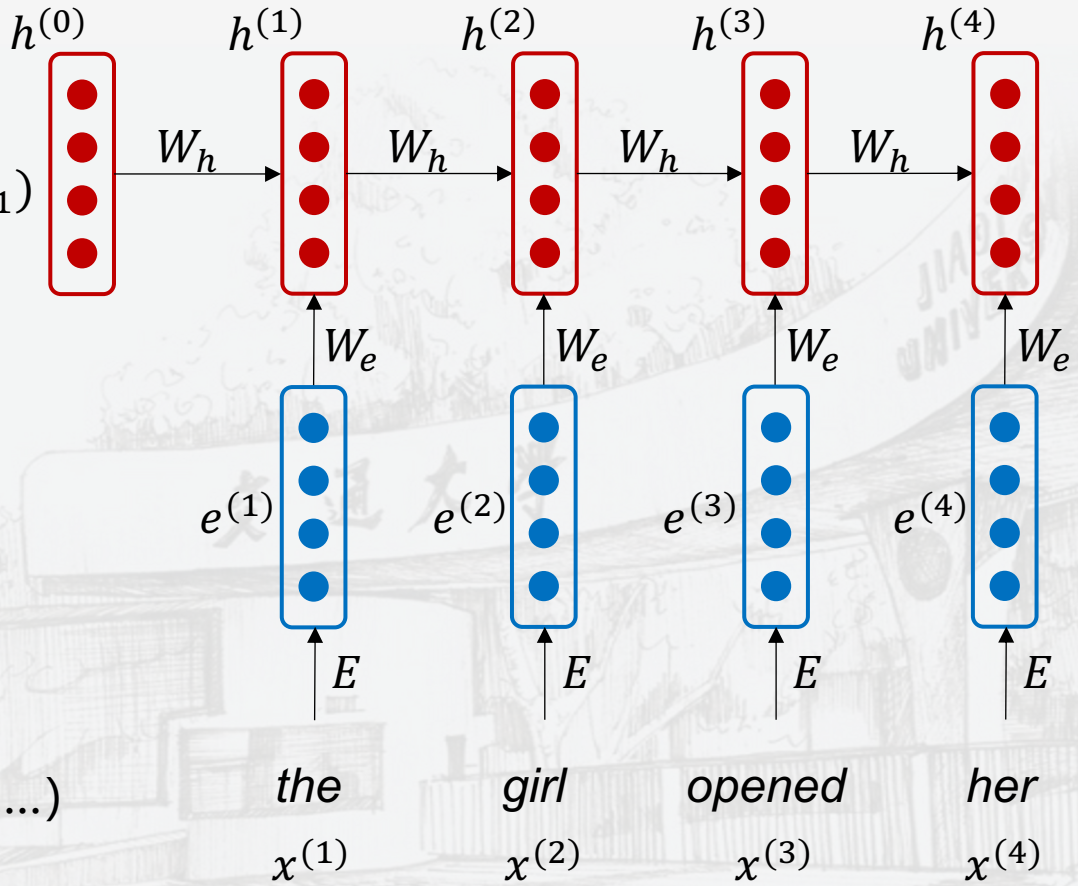
$h^{(0)}$ 为初始隐藏状态

嵌入式词向量

$e^{(t)} = Ex^{(t)}$

词向量（one-hot、分布式表示 ……）

$x^{(t)} \in \mathbb{R}^{|V|}$

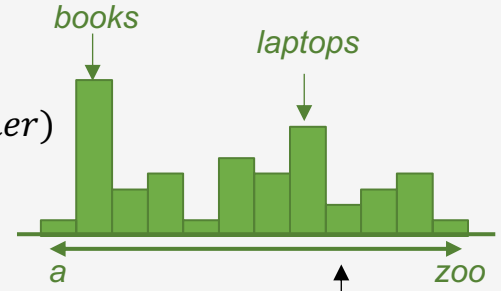


基于递归神经网络的LM

输出层

$\hat{y}^{(t)} = softmax(Uh^{(t)} + b_2) \in \mathbb{R}^{|V|}$

$\hat{y}^{(4)} = P(x^{(5)}|the\ girl\ opened\ her)$



隐藏层

$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e^{(t)} + b_1)$

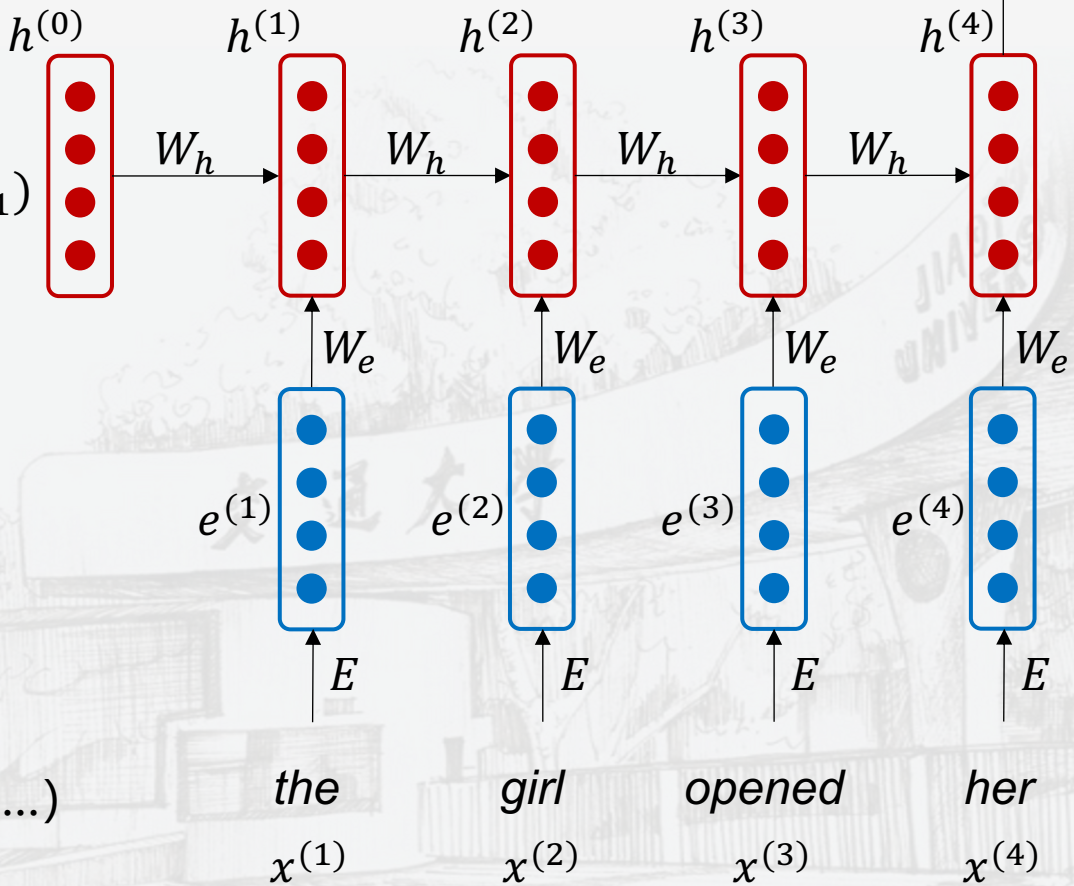
$h^{(0)}$ 为初始隐藏状态

嵌入式词向量

$e^{(t)} = Ex^{(t)}$

词向量（one-hot、分布式表示 .....）

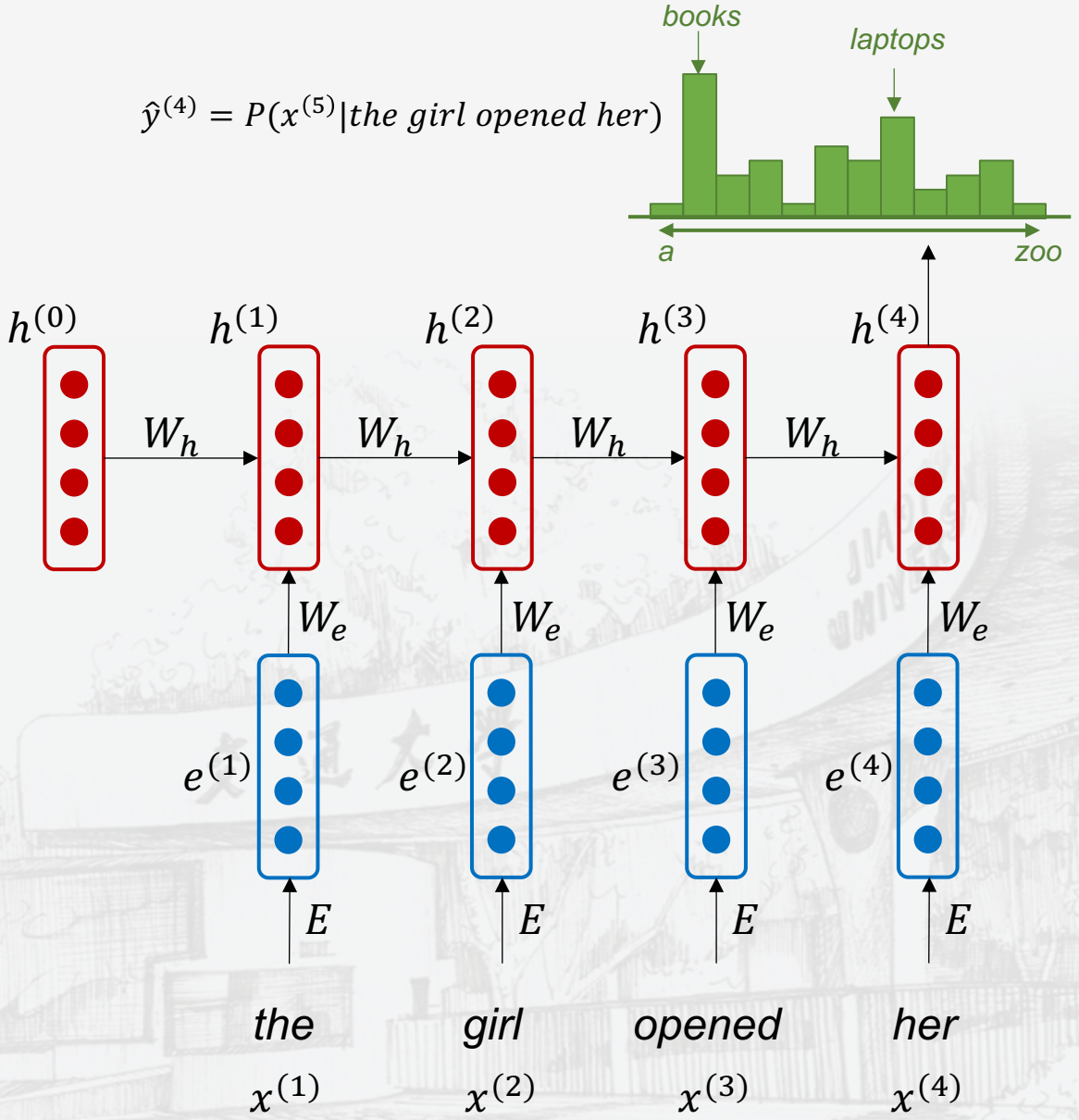
$x^{(t)} \in \mathbb{R}^{|V|}$



基于递归神经网络的LM

优点

- 可处理任意长度句子；
- 第 t 步的计算（理论上）使用了前面多步的信息；
- 模型体量不随着输入变长而增加；
- 每一步使用同一个 W ，降低计算量。



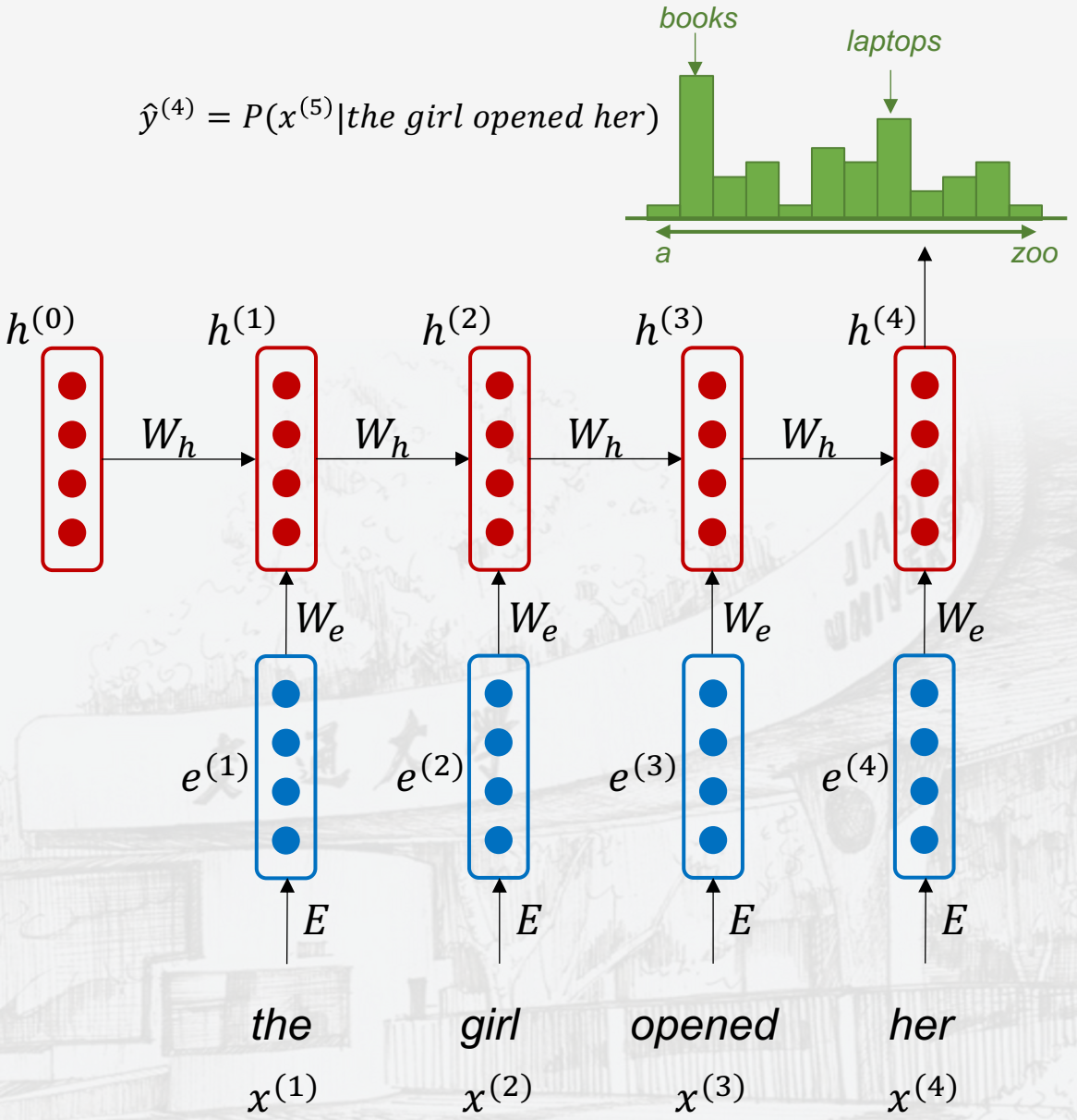
基于递归神经网络的LM

优点

- 可处理任意长度句子；
- 第 t 步的计算（理论上）使用了前面多步的信息；
- 模型体量不随着输入变长而增加；
- 每一步使用同一个 W ，降低计算量。

缺点

- 递归计算缓慢；
- 实际上，将前面很多步的信息完整传递是困难的。



基于递归神经网络的LM

训练

1. 获取一个大的语料库， 词序列形式 $x^{(1)}$，$x^{(2)}$,…，$x^{(t)}$ （其他格式要预处理）；

基于递归神经网络的LM

训练

1. 获取一个大的语料库， 词序列形式 $x^{(1)}$，$x^{(2)}$,…，$x^{(t)}$ （其他格式要预处理）；
2. 输入 RNN；计算每一步 t 的输出概率 $\hat{y}^{(t)}$； （也就是每一个词）

基于递归神经网络的LM

训练

1. 获取一个大的语料库， 词序列形式 $x^{(1)}$，$x^{(2)}$,…，$x^{(t)}$ （其他格式要预处理）；
2. 输入 RNN；计算每一步 t 的输出概率 $\hat{y}^{(t)}$； （也就是每一个词）
3. 使用预测词和真实词的交叉熵做为损失函数；

$$J^{(t)}(\theta) = CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{w \in V} \boldsymbol{y}_w^{(t)} \log \hat{\boldsymbol{y}}_w^{(t)} = -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

基于递归神经网络的LM

训练

1. 获取一个大的语料库， 词序列形式 $x^{(1)}$，$x^{(2)}$，…，$x^{(t)}$ （其他格式要预处理）；
2. 输入 RNN；计算每一步 t 的输出概率 $\hat{y}^{(t)}$； （也就是每一个词）
3. 使用预测词和真实词的交叉熵做为损失函数；

$$J^{(t)}(\theta) = CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{w \in V} \boldsymbol{y}_w^{(t)} \log \hat{\boldsymbol{y}}_w^{(t)} = -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

4. 计算整个训练集的损失；

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) = \frac{1}{T}\sum_{t=1}^{T} -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

西安交通大学
XI'AN JIAOTONG UNIVERSITY

Q & A