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

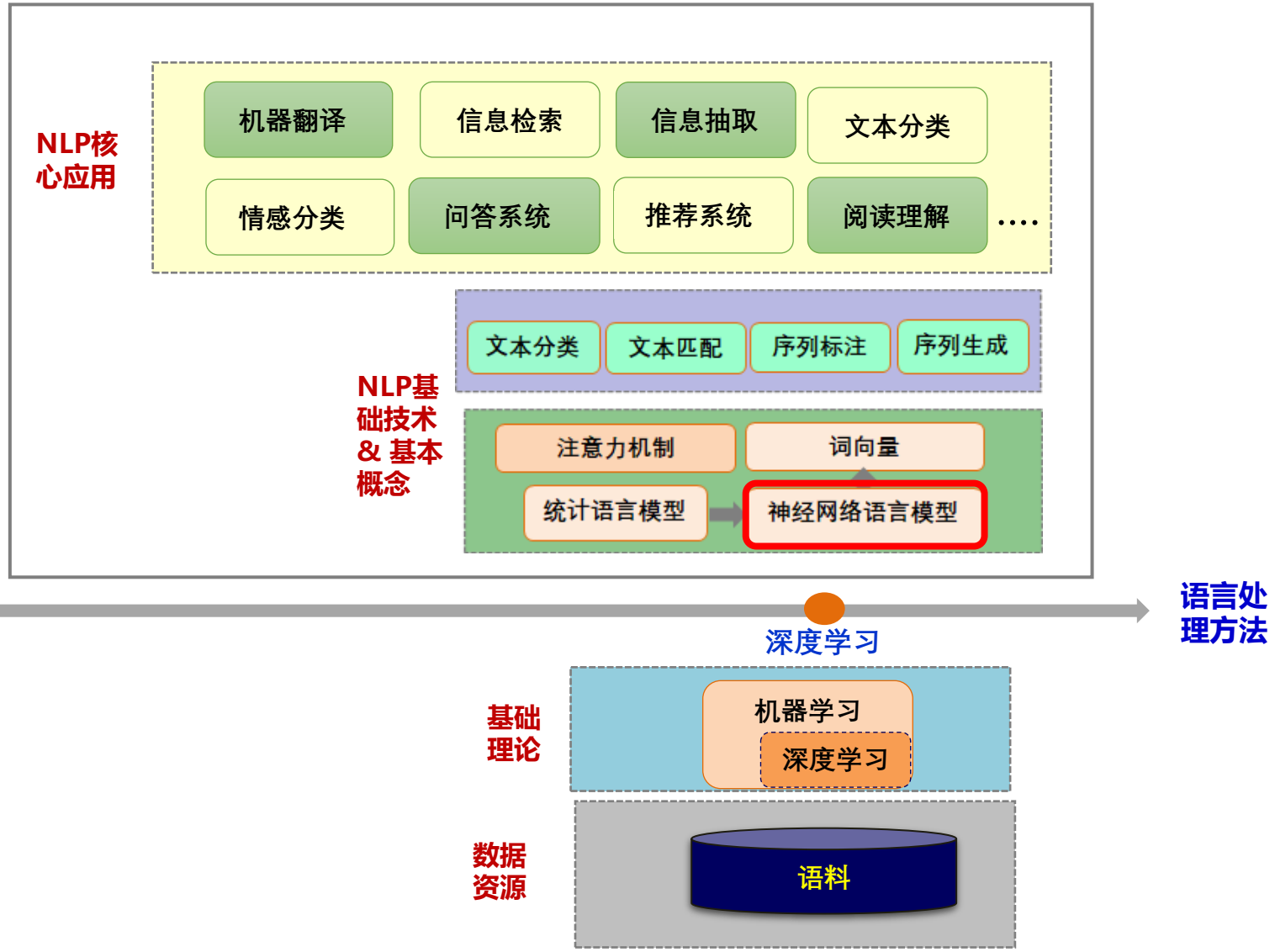
第 8 章 神经网络语言模型

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：

1. 介绍神经网络语言模型的基本概念 2.介绍 NNLM神经网络语言模型结构和训练方法 3. 介绍RNNLM神经网络语言型结构和训练方法以及RNNLM模型的变形

本章教学目的：

了解并掌握神经网络语言模型的概念，模型结构及训练方法

内 容 提 要

8.1 概述

8.2 NNLM 模型

8.3 RNNLM 模型

8.1 概述

统计语言模型回顾

用句子S=$w_1,w_2,...,w_n$ 的 **概率** p(S) 来定量的刻画句子。

$$p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$$

输入： 句子 S

输出： 句子概率p(S)

参数： $p(w_i|w_1,...,w_{i-1})$

统计语言模型：
用概率统计法学习参数

说明：

(1) w_i 可以是字、词、短语或词类等等，称为统计基元。通常以“词”代之。

(2) w_i 的概率由 w_1, ..., w_{i-1} 决定，由特定的一组 w_1, ..., w_{i-1} 构成的一个序列，称为 w_i 的历史（history）。

8.1 概述

问题1： 由于受参数规模限制，需对对词的历史信息进行简化，一般假设一个词的出现概率只与它前面的*n*-1个词相关，距离大于等于n的上文词会被忽略（n-gram）

根据对 p(wi|w1,...,wi-1)的简化程度而定义：

- 1 元模型（unigram）：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i)$ w_i 独立于历史
- 2 元模型（bigram）：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i|w_{i-1})$ w_i 保留前1个词序
- 3 元模型（trigram）：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(wi|w_{i-2}, w_{i-1})$ w_i 保留前2个词序
- ……
- n 元模型（n-gram）：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(wi|w_{i-(n-1)} \ldots w_{i-1})$ w_i 保留前n个词序

8.1 概述

问题2： n-gram 语言模型参数 $p(w_i|w_{i-(n-1)} \ldots w_{i-1})$ 由最**大似然估计**求得使得：

- **存在数据稀疏问题，需要数据平滑。**而平滑技术错综复杂而且需要回退到底阶，使得　该模型无法面向更大的n元文法获取更多的词序信息
- **基于最大似然估计的语言模型缺乏对上下文的泛化**

$$p(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{\sum_{w_i} c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i-1})}$$

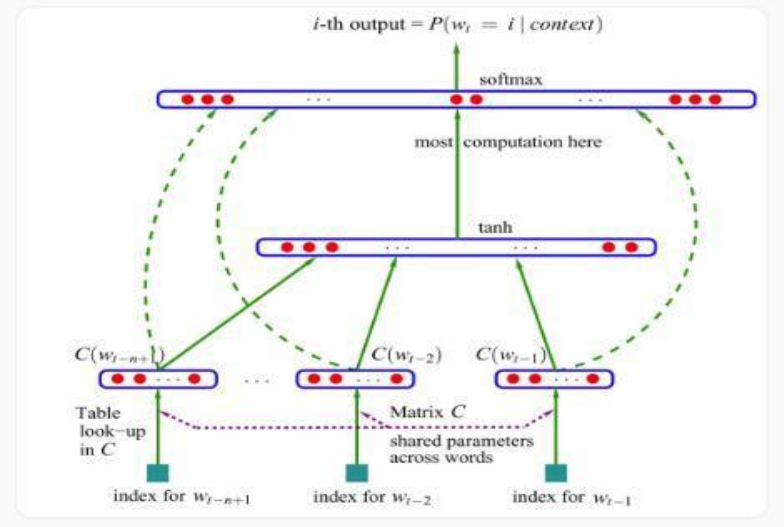
其中：$\sum_{w_i} c(w_{i-n+1}^{i-1})$ 是历史串 w_{i-n+1}^{i-1} 在给定语料中出现的次数

$\sum_{w_i} c(w_{i-n+1}^{i})$，为 w_{i-n+1}^{i-1} 与 w_i 同现的次数。

神经网络语言模型提出

2001年 - 神经语言模型（Neural language models）— 里程碑

Bengio 等人2001年提出了第一个神经网络语言模型 (Neural language models) 它是一种前馈神经网络，该模型在学习语言模型的同时，也得到了词向量。



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

8.1 概述

神经网络语言模型概念

用句子$S=w_1,w_2,\ldots,w_n$ 的 **概率** p(S) 来定量的刻画句子。

$$p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$$

输入： 句子 S

输出： 句子概率p(S)

参数： $p(w_i|w_1,\ldots,w_{i-1})$

统计语言模型：
用概率统计法学习参数

神经网络语言模型：
用神经网络学习参数

根据所用神经网络不同，分为：

- **NNLM 模型 （使用DNN）**
- **RNNLM 模型 （使用RNN）**

内 容 提 要

8.1 概述

8.2 NNLM 模型

8.3 RNNLM 模型

8.2 NNLM模型

目标： 用神经网络DNN 求语言模型 $p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$

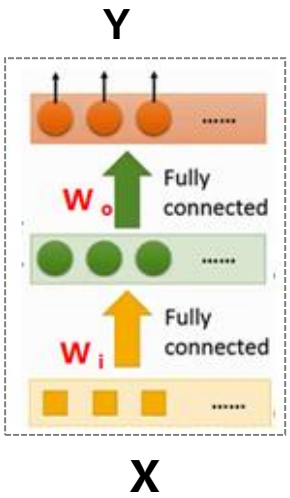
模型参数 $p(w_i|w_{i-(n-1)} \ldots w_{i-1})$

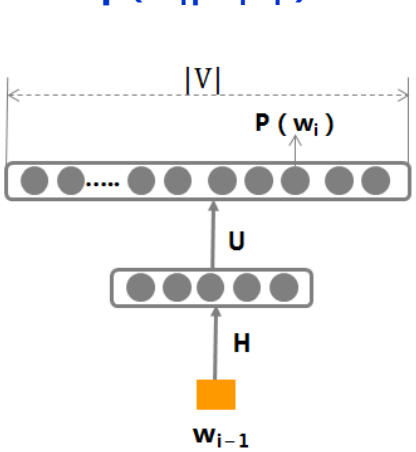
■ **2 元文法模型（bigram）**：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i|w_{i-1})$ w_i 保留前1个词序

语句 s = w_1 w_2 … w_n 的 概率p(S) 为：

$$p(s) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_2) \times \ldots \times p(w_n|w_{n-1})$$

DNN



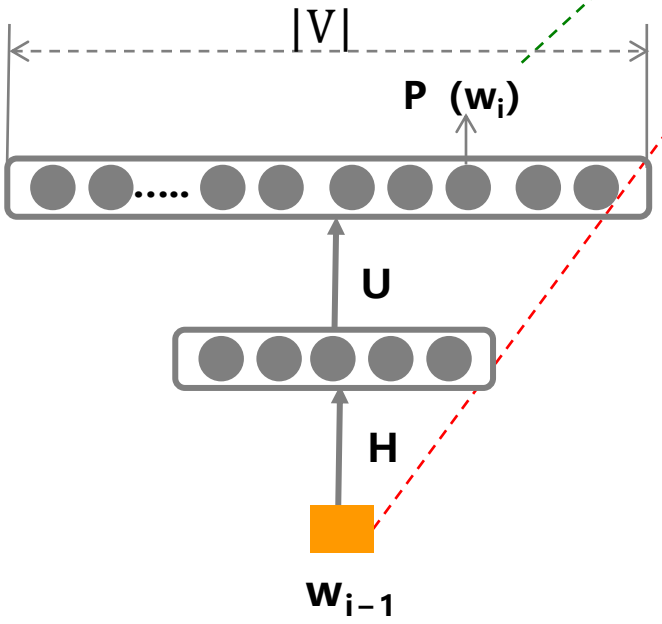


8.2 NNLM模型

（1）NNLM模型结构

语言模型参数

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



softmax(y)

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

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

隐藏层: $h=\tanh(XH+b^1)$

输入层: X: 词 w_{i-1}

参数: $\theta=\{H, U, b^1, b^2\}$

神经网络参数

输出层有|V|个元素，V是有限词表包括未登录词标识UNK和句子开始和结束补齐符号，一般在10000≈1000000左右，常见规模 70000左右

8.2 NNLM模型

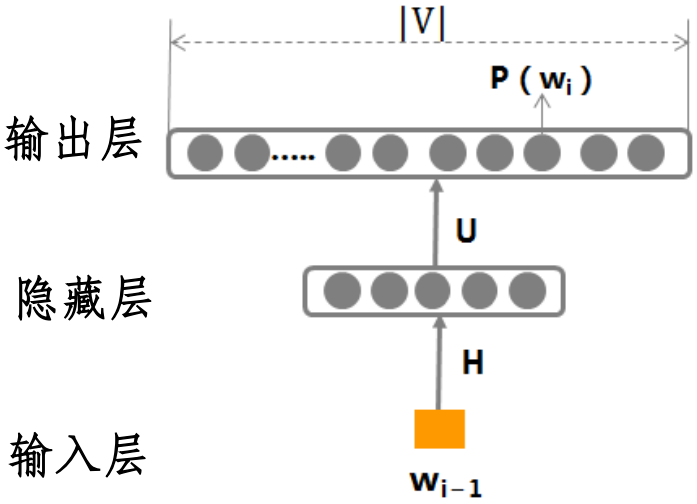
(2) NNLM模型学习 (2-gram)

参数: $\theta = \{ H, U, b^1, b^2 \}$

输出: $p(w_i | w_{i-1})$

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

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



- **语料:** (“无监督”)

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

实例: X: 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:

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

- **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

$$\theta \leftarrow \theta + \alpha \frac{\partial \log P(w_i | w_{i-1})}{\partial \theta}$$

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

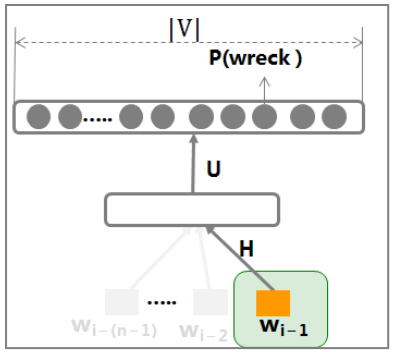
8.2 NNLM模型

(3) NNLM模型预测

例: P(“wreck a nice beach”)

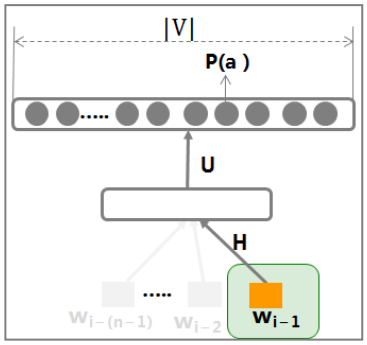
=P(wreck | START)P(a | wreck)P(nice | a)P(beach | nice)

P(wreck | START)



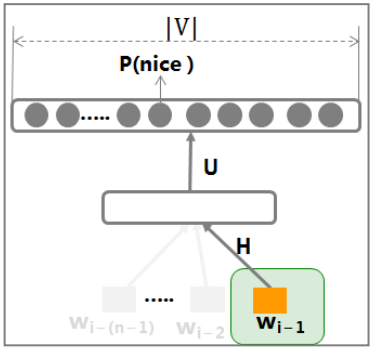
“START”

P(a | wreck)



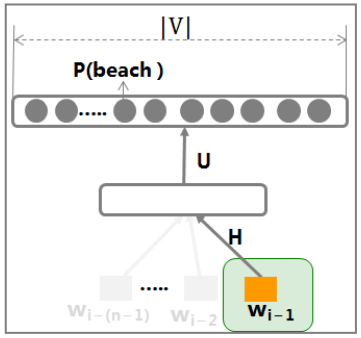
“wreck”

P(nice | a)



“a”

P(beach | nice)

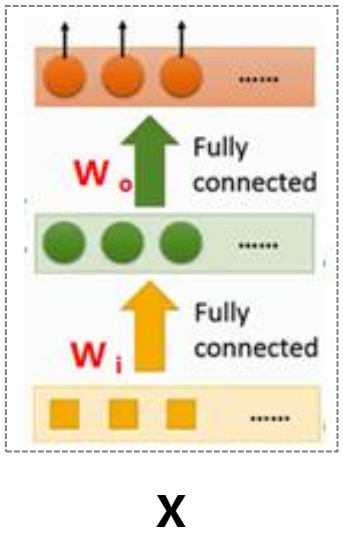


“nice”

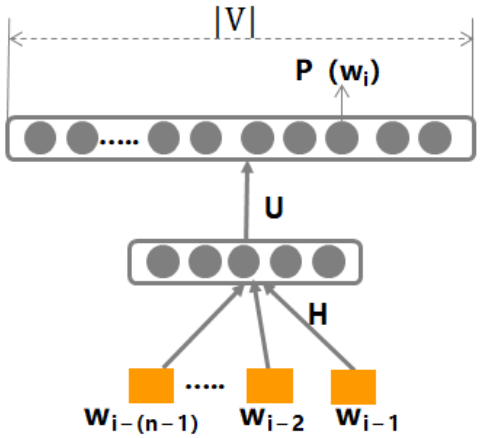
8.2 NNLM模型

- **n 元文法模型（n-gram）**：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i|w_{i-(n-1)}\ \ldots\ w_{i-1})$ w_i 保留前n个词序

DNN



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

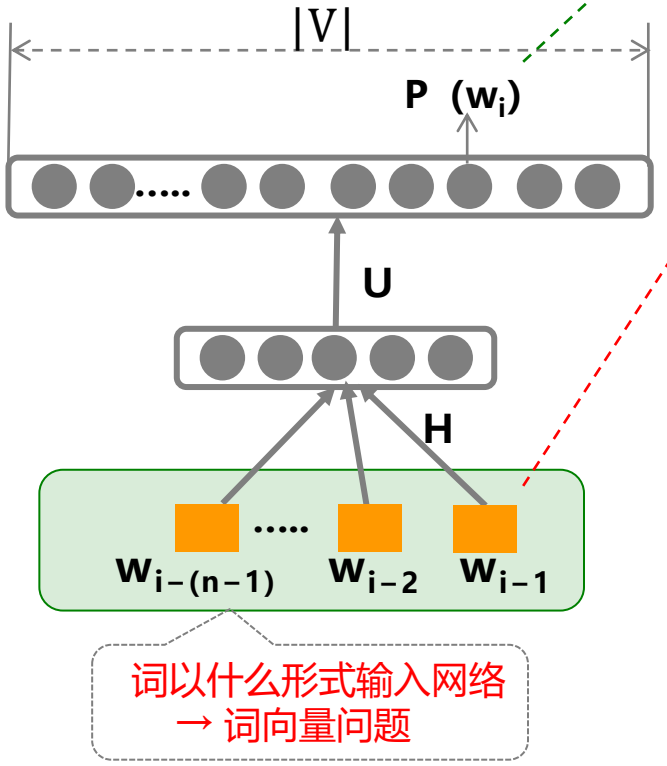


8.2 NNLM模型

(1) NNLM模型结构

语言模型参数

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



softmax(y)

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

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

隐藏层: $h=\tanh(XH+b^1)$

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

参数: $\theta=\{H,U,b^1,b^2\}$

神经网络参数

8.2 NNLM模型

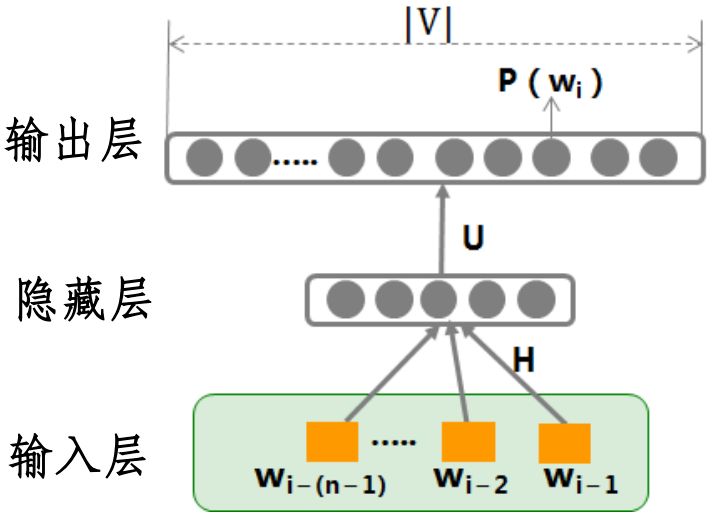
(2) NNLM模型学习 (n-gram)

参数: $\theta = \{ H, U, b^1, b^2 \}$

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

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

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



- **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:

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

- **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

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

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

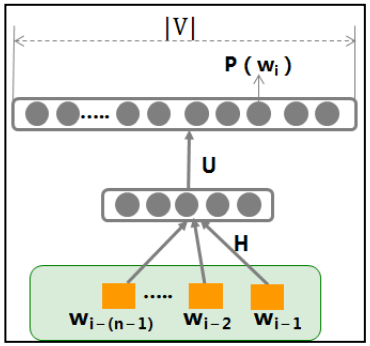
8.2 NNLM模型

(3) NNLM模型预测 (n-gram) 如 n=4

例: P(“wreck a very nice beach”)

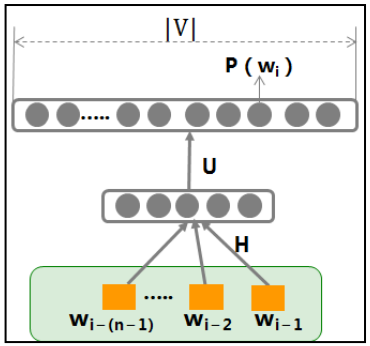
=P(very | START wreck a)P(nice | wreck a very) P(beach | a very nice)

P(very | START wreck a)



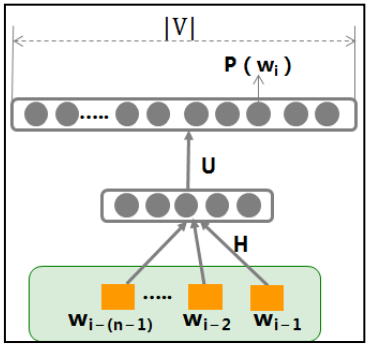
“START wreck a ”

P(nice | wreck a very)



“wreck a very ”

P(beach | a very nice)



“a very nice ”

每求一个参数用一遍神经网络

内 容 提 要

8.1 概述

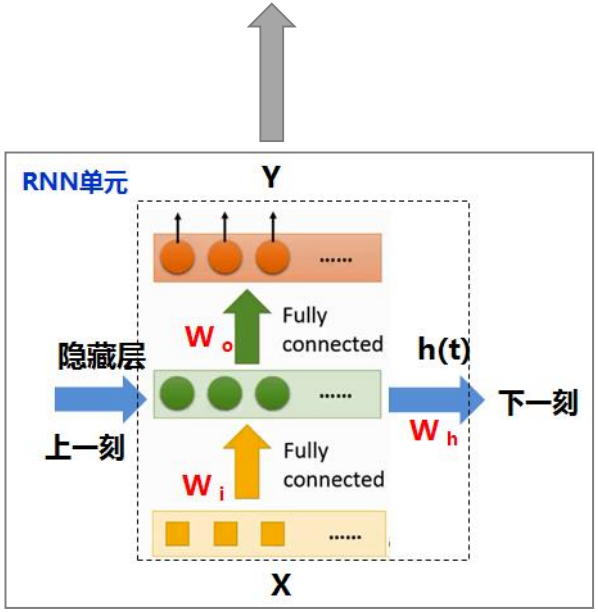
8.2 NNLM 模型

8.3 RNNLM 模型

8.2 NNLM模型

目标： 用循环神经网络RNN 求语言模型 $p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$

模型参数 $p(w_i | w_{i-(n-1)} \cdots w_{i-1})$

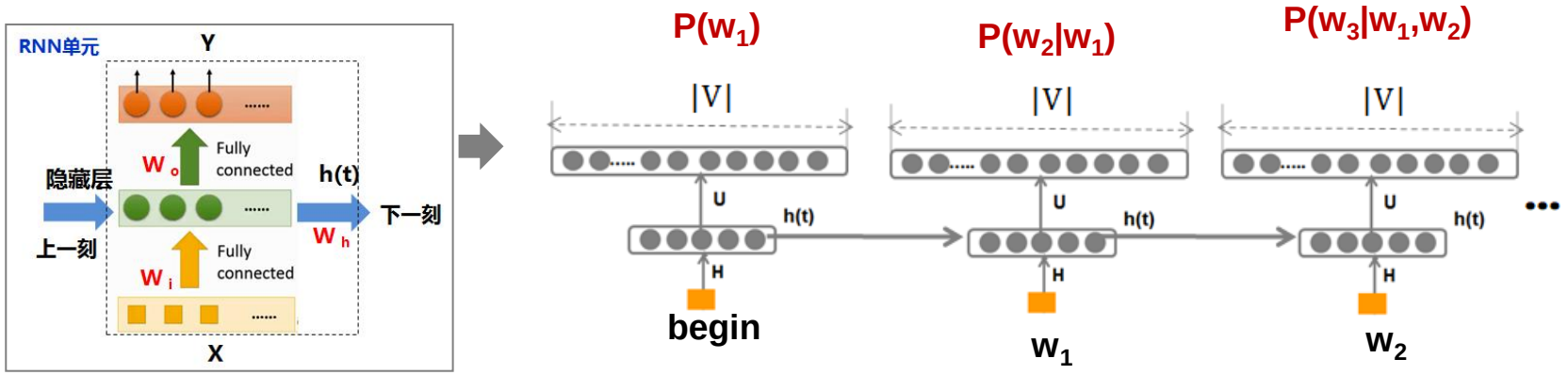


8.3 RNNLM 模型

■ **2 元文法模型（bigram）**：$p(w_1,\ldots,w_m)=\prod_{i=1}^{m} p(w_i|w_{i-1})$ w_i 保留前1个词序

语句 s = w_1 w_2 … w_n 的 概率p(S)定义为：

$$p(s) = p(w_1)\times p(w_2|w_1)\times p(w_3|w_2)\times \ldots \times p(w_n|w_{n\text{-}1})$$

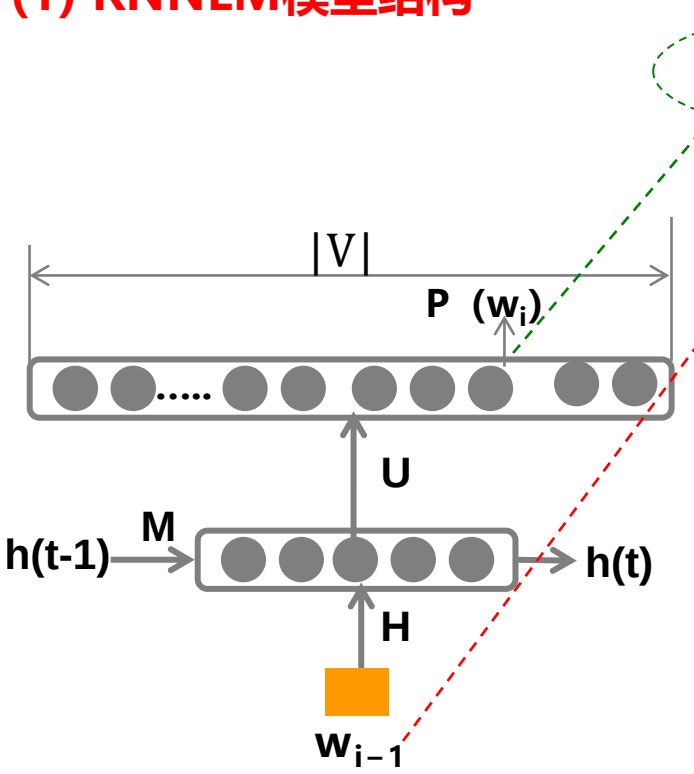


随着模型逐个读入语料中的词w_1;w_2 ….隐藏层不断地更新为h(1),h(2)…..,通过这种迭代推进方式，每个隐藏层实际上包含了此前所有上文的信息，相比NNLM 只能采用上文n 元短语作为近似，RNNLM 包含了更丰富的上文信息，也有潜力达到更好的效果。

8.3 RNNLM 模型

(1) RNNLM模型结构

语言模型参数



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： $h(t) = \tanh(XH + Mh(t-1) + b^1)$

输入层： X: 词 w_{i-1}

参数： $\theta = \{H, U, M, b^1, b^2\}$

神经网络参数

8.3 RNNLM 模型

(2) RNNLM模型学习

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

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

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

- **语料：（“无监督”）**

文本： $S = w_1, w_2, \ldots, w_n$,

……

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

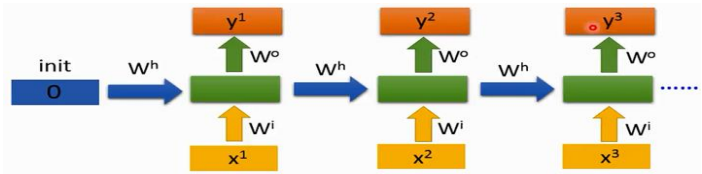
$\hat{Y}$ ： $w_1, w_2, \ldots, w_{n-1}\ w_n$

- **目标函数：**

对于整个语料，语言模型需要最大化

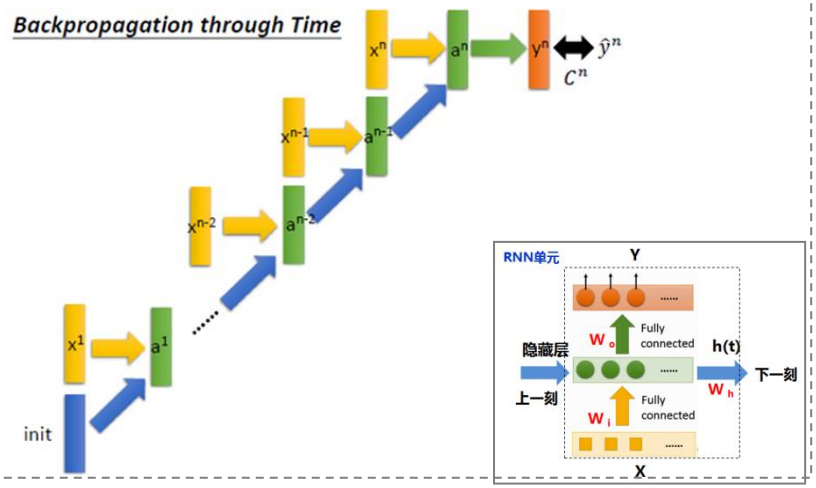
$$\sum_{w_{i-1}} \sum_{i \in D} \log P(w_i | w_{i-1})$$

RNN网络



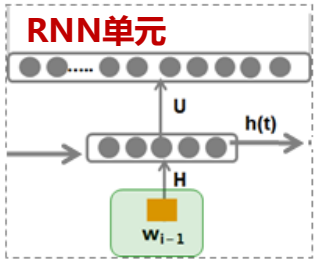
- **参数训练：**

(BPTT) 随机梯度下降法优化训练目标:



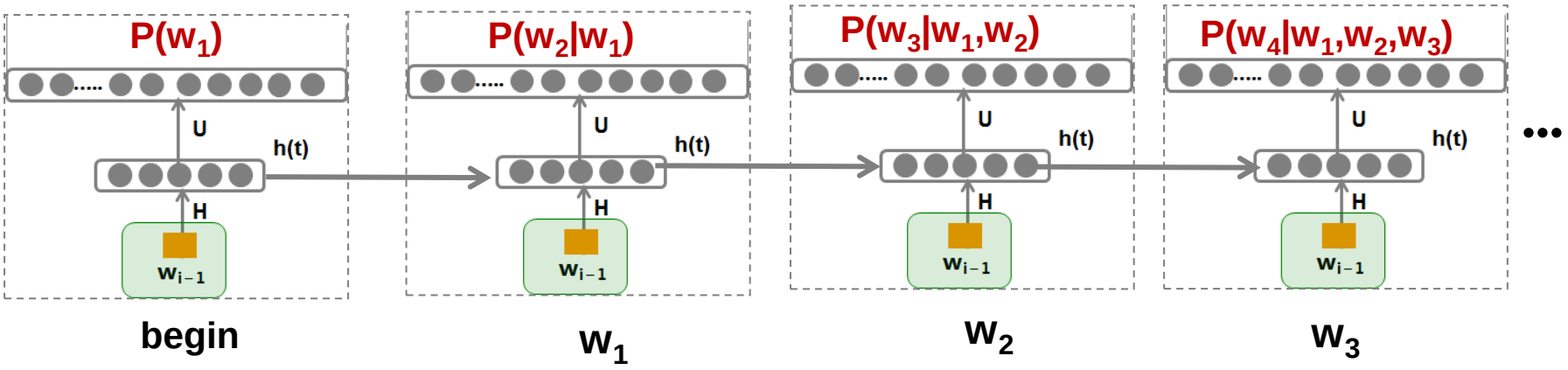
8.3 RNNLM 模型

(3) RNNLM模型预测



例：　$P(w_1, w_2, w_3, \cdots, w_n)$

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



8.3 RNNLM 模型(优点)

RNNLM 优点:

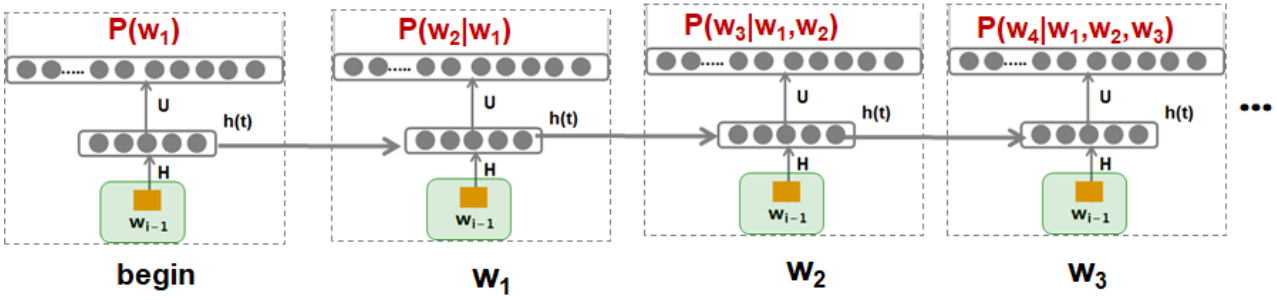
- **模型（RNNLM）可以保留每个词的全部历史信息，不需简化为n-gram**
- **引入词向量作为输入后不需要数据平滑**

在神经网络中常用RNN语言模型

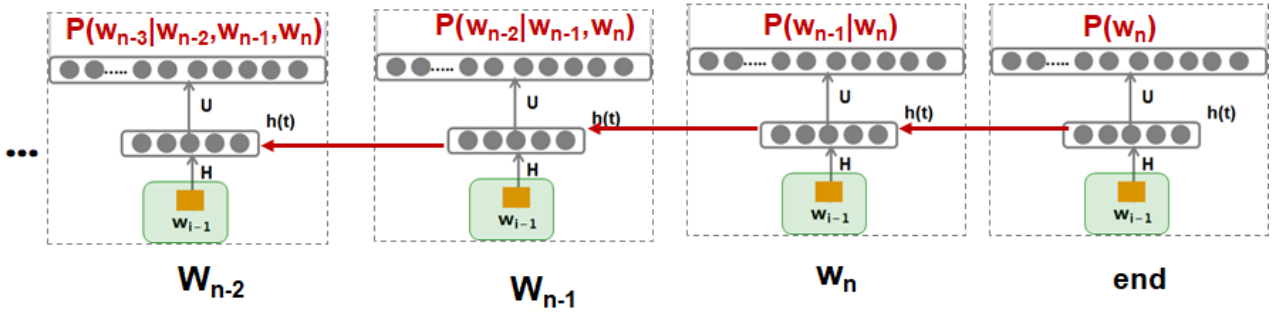
8.3 RNNLM 模型(变形)

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

■ 正向语言模型

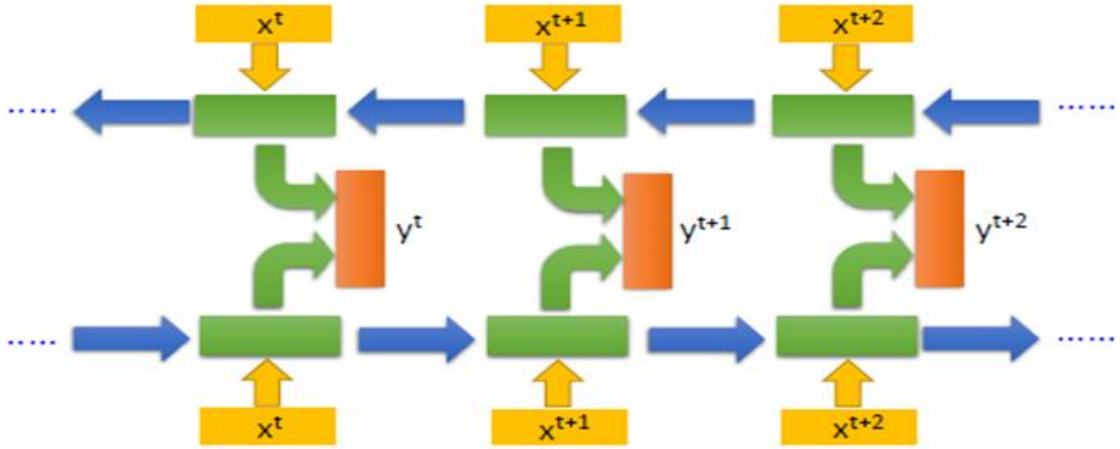


■ 反向语言模型



8.3 RNNLM 模型(变形)

■ 双向语言模型



$$\vec{h}_t = f(\vec{W}x_t + \vec{V}\vec{h}_{t-1} + \vec{b})$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})$$

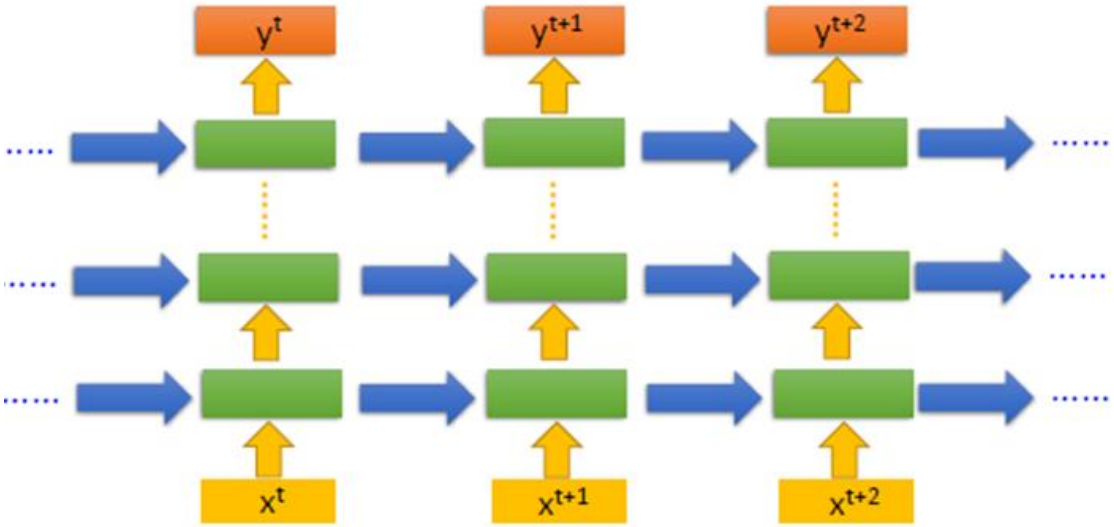
$$y_t = g(U[\vec{h}_t; \overleftarrow{h}_t] + c)$$

每个时刻都有一个正向输入的隐层 $\overrightarrow{ht}$ 和·一个反向输入隐层 $\overleftarrow{h_t}$

两个隐层分别可以表示一个词的上文信息和下文信息

8.3 RNNLM 模型(变形)

■ 单向多层RNN语言模型



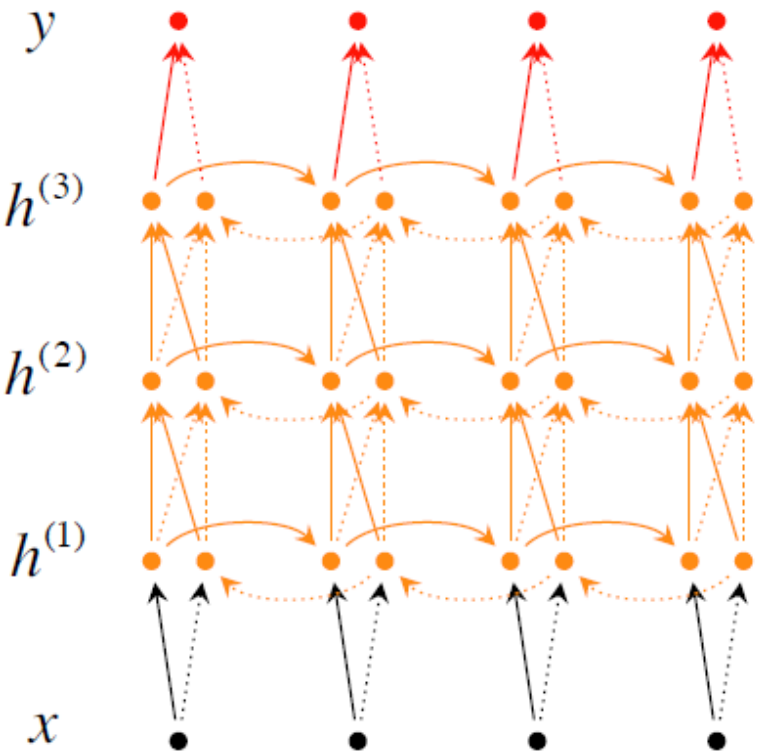
$$h^i(t) = \sigma(W^i{}_i\, h^{i-1}(t) + W^i{}_h\, h^i(t-1) + b^i)$$

$$Y = \text{softmax}(W_o h^L(t))$$

采用多个隐层，每个隐层向后一层传递序列信息

8.3 RNNLM 模型(变形)

■ **双向多层RNN语言模型**



$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)} h_t^{(i-1)} + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)})$$

$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\overrightarrow{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

各层每个时刻都有一个正向输入 $\overrightarrow{h^i}_t$ 和·一个反向输入 $\overleftarrow{h^i}_t$

每个隐层向后一层传递序列信息

谢谢各位！

Q&A