5-1 Recurrent Neural Network I

Zhonglei Wang

WISE and SOE, XMU, 2025

Contents

研究动机

1. 自然语言处理（Natural Language Processing，NLP）中的常见问题
   - 机器翻译
   - 情感分析
   - 聊天模型（Chat Generative Pre-trained Transformer，ChatGPT）
2. 例如，寻找下面句子中的目的地（本章以英文句子为例）
   - I arrive at Beijing from Xiamen
   - I leave Beijing to Xiamen
3. 问题
   - 如何将一个句子转化为可用于计算的特征?
   - 适合自然语言处理的模型是什么?

研究动机

1．在自然语言处理的问题中，我们通常采取如下步骤

- 词元化（Tokenization）
- 词嵌入（Embedding）
- 建模

(为了提取特性)

(为了分析)

词元化

1．基于词典（vocabulary）的词元化的难点

- 例如名字等，有些词汇并不会出现在词典里
- 对处理标点符号 “,.;:?!” 等没有明确的结论
- 对于同一个词源的不同（英语等）单词，需要不同的词元化
  - 例如，“walk, walks, walked”...

2．可能的解决方案：寻找一个词典，能够包括如下信息

- 常见单词
- 允许由单词（及其前后缀）形成的词组的出现

词元化

1．对于英语等，词元化是非常容易做到的

2．例如，当我们考虑 “I arrive at Beijing from Xiamen.”

3．我们可以将其分解成七个词元

I / arrive / at / Beijing / from / Xiamen / .

4．得到词元后，我们给每个词元分配一个词向量

（由数字组成的具有固定长度的列向量）

独热编码

1．在处理实际问题时，一个词典的大小约为$N(\approx 30,000)$

2．独热编码可被用于为字典中的每个词元进行赋值

- 每个词元对应的向量长度为N
- 该向量除了一个位置为 1 外，其他元素均为 0；1 的位置代表该词元在字典中的位置

关于独热编码的说明

1．缺点

- 高维且稀疏：编码中绝大多数信息无用，尤其当字典规模很大时
- 对新次元的泛化能力较弱
- 由于词向量维度高，容易产生过拟合
- 不能用来对“关系”进行建模，例如“Man-Women”
- 对字典中的微小变化较为敏感：增减词汇将改变整个编码系统
- ……

词嵌入

1. 我们希望得到一种新的为词元赋值词向量的方式–词嵌入（Word embedding）
   - 词向量维度较低($\approx 1,024$)
   - 反映词元之间的逻辑关系
2. 一些可行的解决方案
   - Word2Vec: CBOW+Skip-gram
   - GloVe: 对 Word2Vec 进行泛化
   - N-Gram: 一个概率模型
   - TF-IDF
   - BERT
   - ……

词嵌入

1．词嵌入是自然语言处理的重要任务

- 词嵌入是基于众多算法学到的结果
- 基于优秀的词嵌入，OpenAI 的 GPT 可以产生更加连贯

  并且符合人们行文逻辑的回复

2．在本章中，我们不讨论产生词嵌入的算法，请自行学习相关内容

3．在接下来的分析中，我们假设我们对每个词元有一个词嵌入

模型

1．任务：寻找下列句子中的目的地

	I	arrive	at	Beijing	from	Xiamen
词嵌入：	$\boldsymbol{x}^{<1>}$	$\boldsymbol{x}^{<2>}$	$\boldsymbol{x}^{<3>}$	$\boldsymbol{x}^{<4>}$	$\boldsymbol{x}^{<5>}$	$\boldsymbol{x}^{<6>}$
标签：	$y^{<1>}=0$	$y^{<2>}=0$	$y^{<3>}=0$	$y^{<4>}=1$	$y^{<5>}=0$	$y^{<6>}=0$

2．什么样的模型适合自然语言处理呢?

模型

1．为什么不能用全连接神经网络或者卷积神经网络?

- 词向量的维度一般很大
- 不同输入（句子）的长度往往不一致
- 特别地，以上两个模型不能处理句子内部不同词元之间的相关关系

循环神经网络（Recurrent Neural Network）

1．一个类似的模型早在 1980 年代就被提出

2．我们以寻找句子中的目的地为例进行讲解

- 本质上讲，这是针对一个句子中的每个词元的二分类问题

3．我们有一个由多个（长度不尽相同的）句子组成的训练集

4．后续我们将讨论其他自然语言处理中的任务

循环神经网络（Recurrent Neural Network）

1．假设我们有一个句子

	I	arrive	at	Beijing	from	Xiamen
词嵌入：	$\boldsymbol{x}^{<1>}$	$\boldsymbol{x}^{<2>}$	$\boldsymbol{x}^{<3>}$	$\boldsymbol{x}^{<4>}$	$\boldsymbol{x}^{<5>}$	$\boldsymbol{x}^{<6>}$
标签：	$y^{<1>}=0$	$y^{<2>}=0$	$y^{<3>}=0$	$y^{<4>}=1$	$y^{<5>}=0$	$y^{<6>}=0$

2．我们需要对该句中的每个词元是目的地的概率$\{\hat{y}^{<i>}:i=1,\ldots,6\}$进行估计

模型模块

1. 初始化$\boldsymbol{a}^{<0>} = 0$

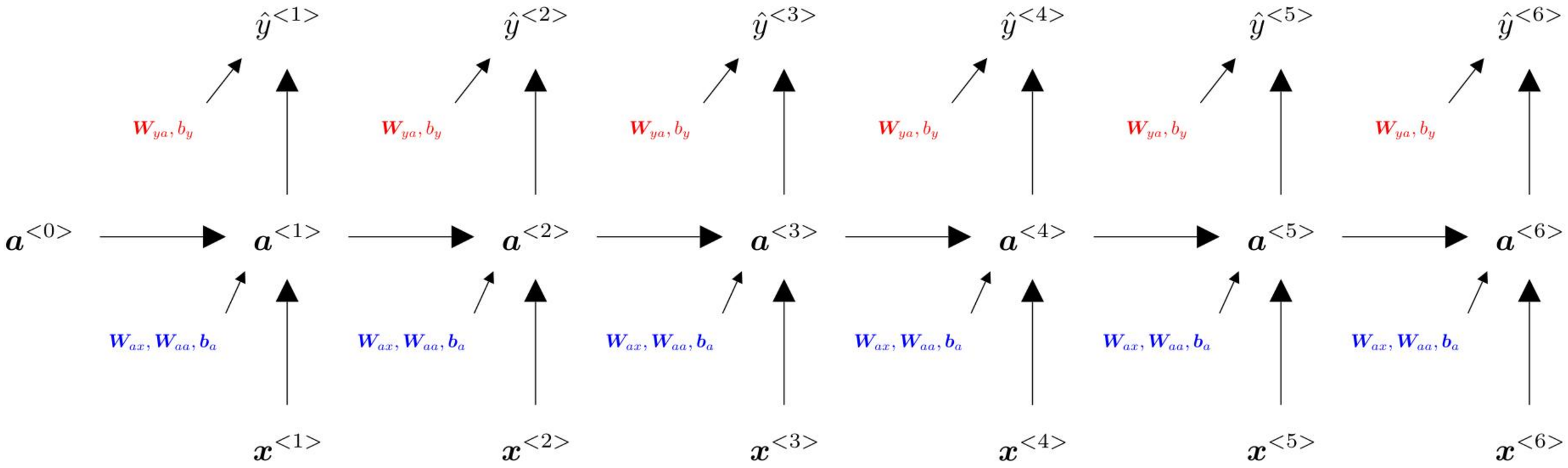
2. 基于当前模型参数$\{\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a, \boldsymbol{W}_{ya}, b_y\}$，我们计算

$$\boldsymbol{a}^{<i>} = \sigma_{ax}(\boldsymbol{W}_{ax}\boldsymbol{x}^{<i>} + \boldsymbol{W}_{aa}\boldsymbol{a}^{<i-1>} + \boldsymbol{b}_a) \quad (i = 1, \ldots, 6)$$

$$\hat{y}^{<i>} = \sigma_{ya}(\boldsymbol{W}_{ya}\boldsymbol{a}^{<i>} + b_y) \quad (i = 1, \ldots, 6)$$

模型模块

$$\hat{y}^{<i>} = \sigma_{ya}(\boldsymbol{W}_{ya}\boldsymbol{a}^{<i>} + b_y) \quad (i = 1, \ldots, 6)$$

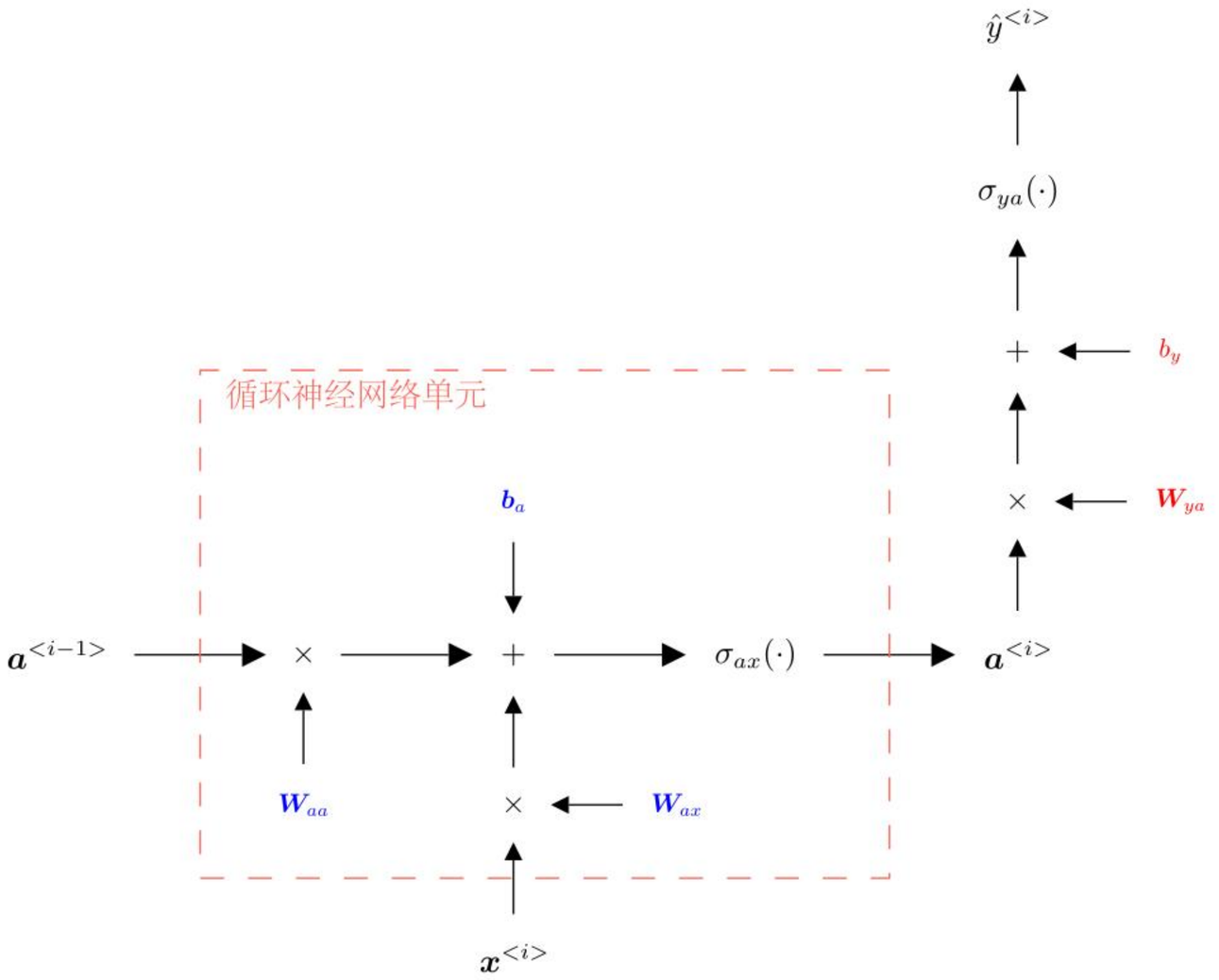


$$\boldsymbol{a}^{<i>} = \sigma_{ax}(\boldsymbol{W}_{ax}\boldsymbol{x}^{<i>} + \boldsymbol{W}_{aa}\boldsymbol{a}^{<i-1>} + \boldsymbol{b}_a) \quad (i = 1, \ldots, 6)$$

说明

1. 由于我们考虑的是二分类问题，代价函数是交叉熵
2. 基于上一页的前向传播过程，我们可以得到对应的后向传播
   - 提示：我们需要将包含模型参数信息的导数都加起来

流程图


$\hat{y}^{<i>}$
$\sigma_{ya}(\cdot)$
b_y
循环神经网络单元
b_a
W_{ya}
$a^{<i-1>}$
$\sigma_{ax}(\cdot)$
$a^{<i>}$
W_{aa}
W_{ax}
$x^{<i>}$

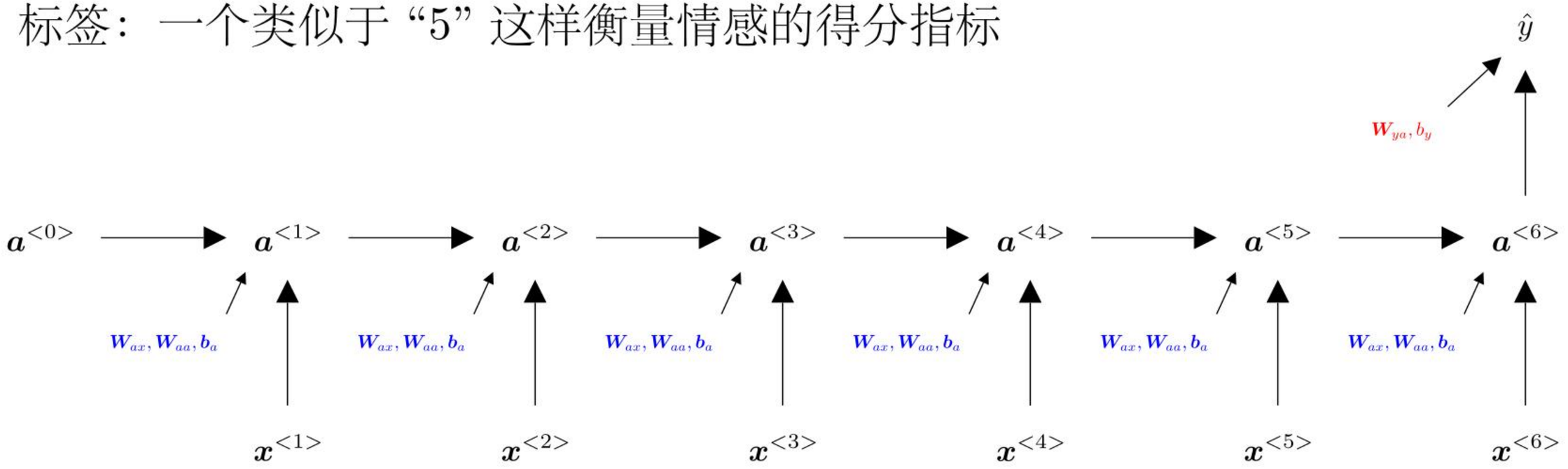
其他例子

1. 情感分析（多对 1）
   - 特征：一个类似于 “I like this movie very much” 的句子
   - 标签：一个类似于 “5” 这样衡量情感的得分指标
2. 机器翻译（多对多）
   - 特征：一个句子
   - 标签：翻译后的句子
3. 文本生成（? 对多）
   - 特征：类似于 “I like ” 的句子的开头部分
   - 标签：基于词补全的句子成分
4. ...

情感分析

1．特征：一个类似于 “I like this movie very much” 的句子

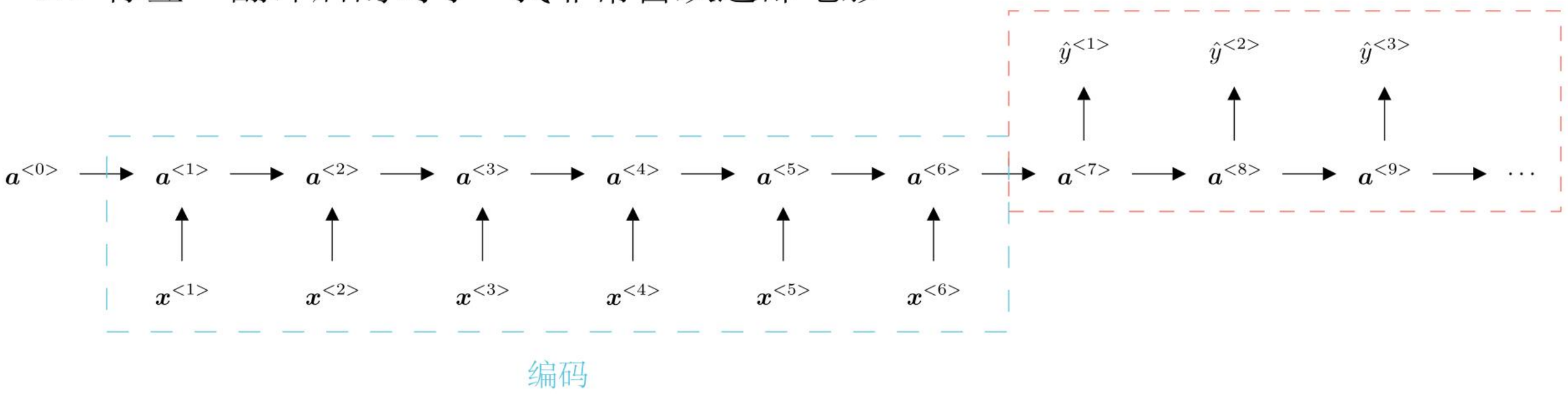
2．标签：一个类似于 “5” 这样衡量情感的得分指标



机器翻译

1．特征：类似于 “I like this movie very much” 的一句话

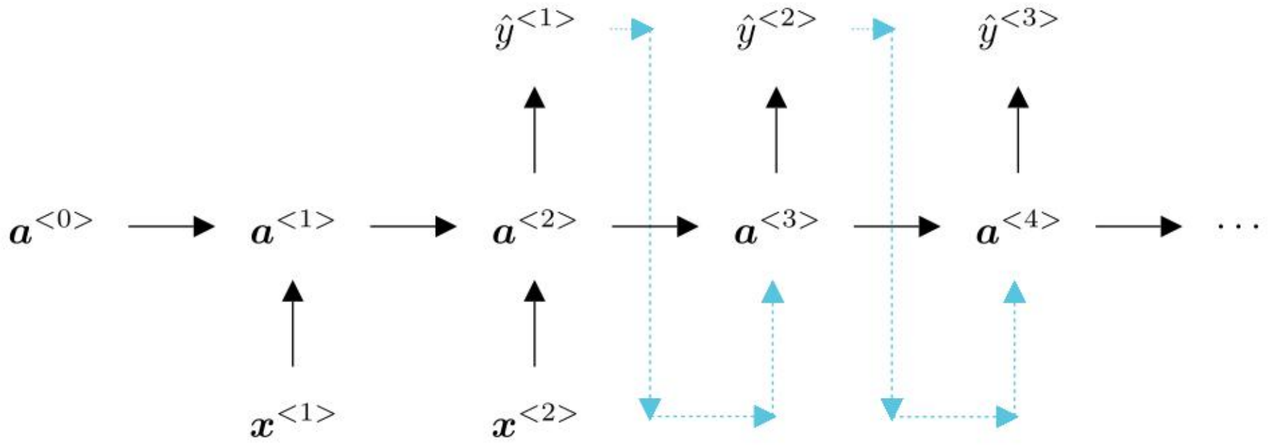
2．标签：翻译后的句子 “我非常喜欢这部电影”



文本生成

1．特征：类似于“I like ”的句子的开头部分

2．标签：基于词补全的句子成分



说明

1. Bengio et al. (1994) 指出 循环神经网络模型不能捕捉长序列依赖关系，原因在于梯度可能消失（概率高）或者爆炸（概率低但后果严重）

2. "This makes gradient-based optimization method struggle, not just because of the variations in gradient magnitudes but because of the effect of long-term dependencies is hidden (being exponentially smaller with respect to sequence length) by the effect of short-term dependencies" (Chung et al., 2014)