



深度学习中的序列模型

及其在 2020 腾讯广告算法大赛中的应用

宋宁宇

麦嘉仪

李卓熹

2020 年 12 月 7 日





C ONTENT 目录

01 | 背景介绍

02 | 方法综述

03 | 比较实验




01

背景介绍



序列模型及其常用场景

序列模型 (sequence model) : 处理语言或者音视频等**前后相互关联**的数据

场景	输入	输出
语音识别		"The quick brown fox jumped over the lazy dog."
音乐生成	无	
情感分析	"There is nothing to like in this movie."	★☆☆☆☆
DNA序列分析	AGCCCCTGTGAGGAACTAG	AG CCCCTGTGAGGAACTAG
机器翻译	Voulez-vous chanter avec moi?	Do you want to sing with me?
视频动作识别		Running
命名实体识别	Yesterday, Harry Potter met Hermione Granger.	Yesterday, Harry Potter met Hermione Granger .



序列数据实例：2020腾讯广告算法大赛

广告受众基础属性预估

数据：

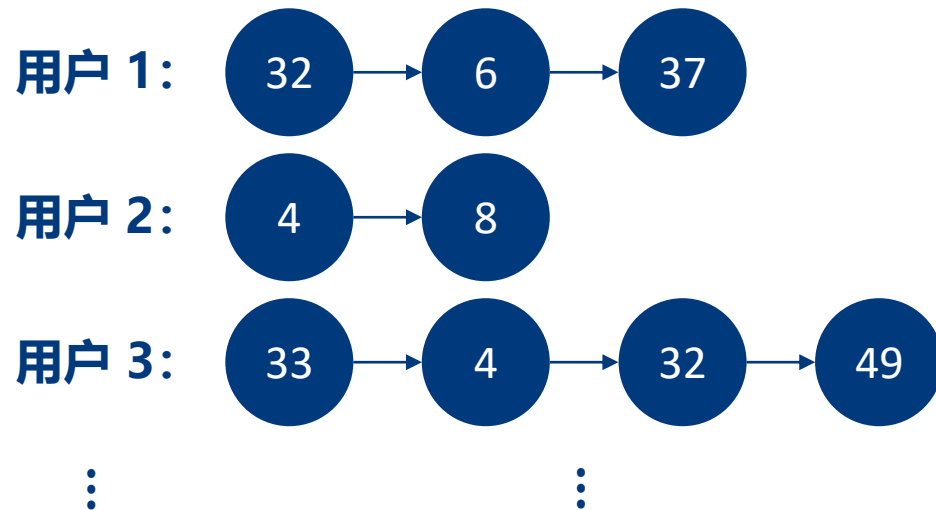
- 用户在 91 天内的 广告素材点击序列
- 用户的 性别 和 年龄（划分为10个区段）
- 训练集有200万用户，测试集有100万用户

目标：根据点击序列，预测用户的性别和年龄

评价指标：年龄 和 性别的 准确率之和

赛题意义：逆向验证广告行业的经典假设；

为填补用户缺失特征提供可能



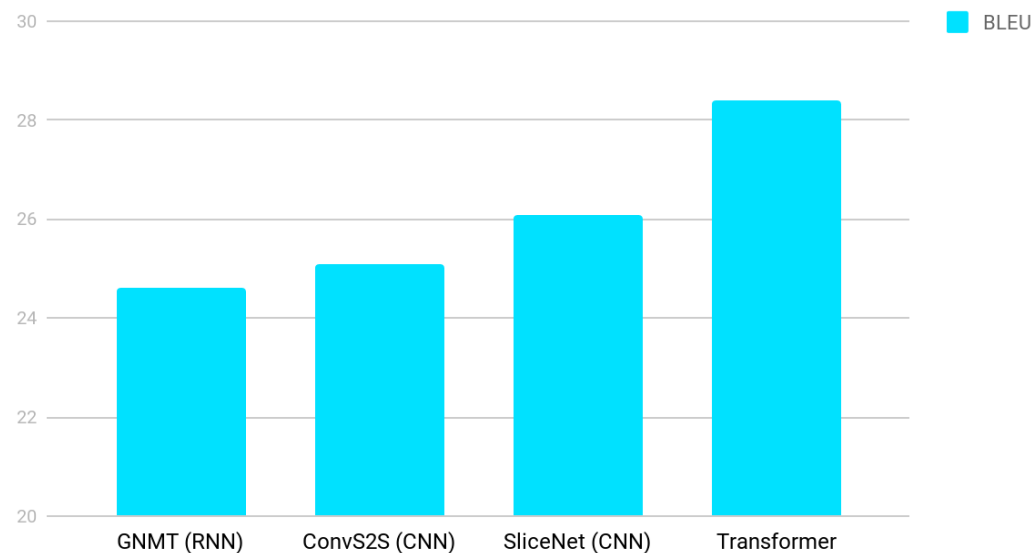
示例：用户所点击的素材id序列

	time	user_id	creative_id	click_times
0	9	30920	567330	1
1	65	30920	3072255	1
2	56	30920	2361327	1
3	6	309204	325532	1

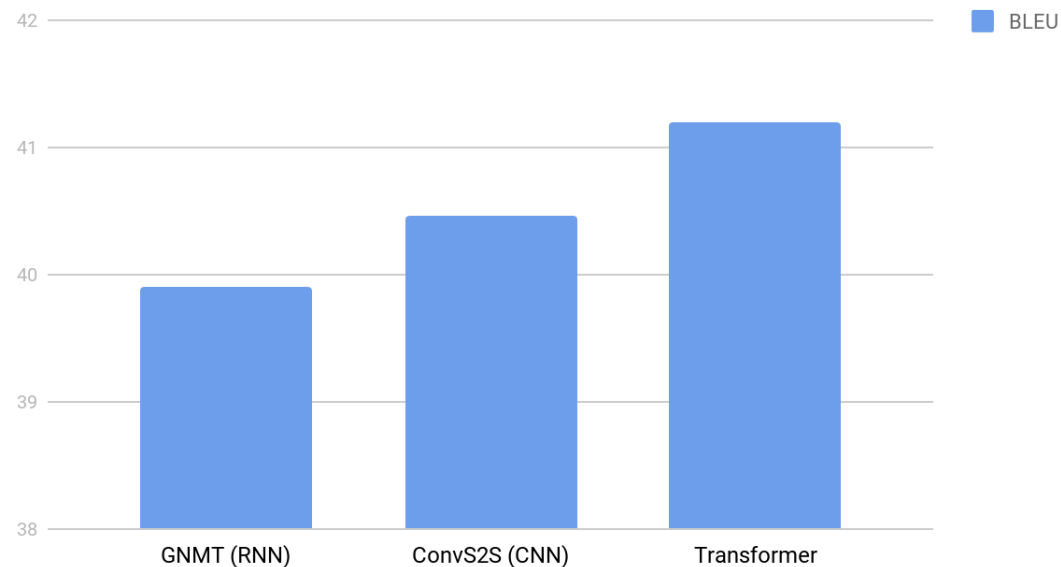
原始数据：用户素材点击

深度学习中的序列模型

English German Translation quality



English French Translation Quality



BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to German/French translation benchmark

近年来，深度学习在序列建模中也有着无可比拟的优势

本次 Project 的目标

- 了解和学习目前比较常用的深度学习序列模型背后的 **直觉** 以及 **基本原理**
- 通过对比实验，探究这些模型在 **实际数据中的表现**
- 探索 **如何更好地利用** 深度学习序列模型

02 | 方法综述

回顾：传统前馈网络

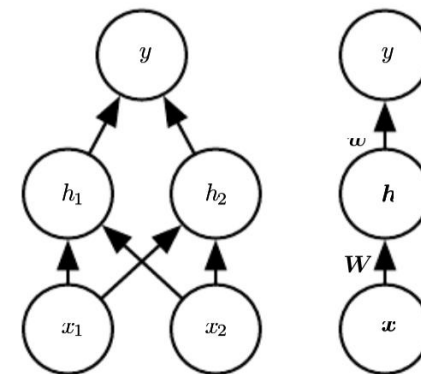
本质：线性变换与非线性变换

- $x \in \mathbb{R}^p \mapsto h \in \mathbb{R}^l : h = f(W^T x)$

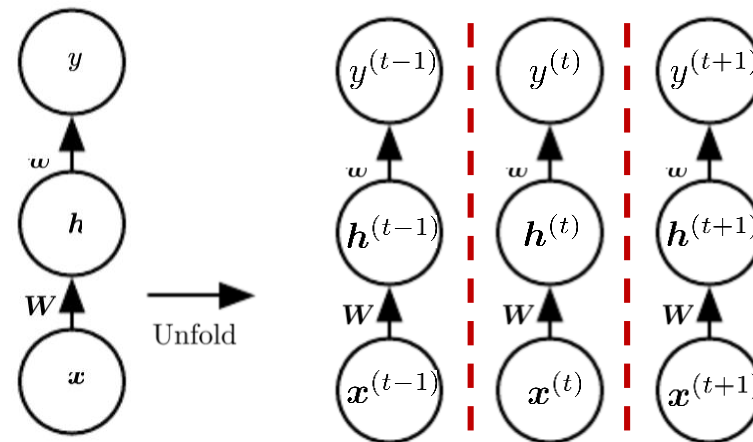
这里 $W^T x$ 是 线性变换 , $f(\cdot)$ 是 非线性变换

为什么不能用于序列建模?

- 考虑总共 T 个时间点的输入序列 $x^{(1)}, x^{(2)}, \dots, x^{(T)}$
- 在处理 x_t 时, 关于 x_{t-1} 的信息 **不会被利用到**



传统前馈网络：两种表示形式

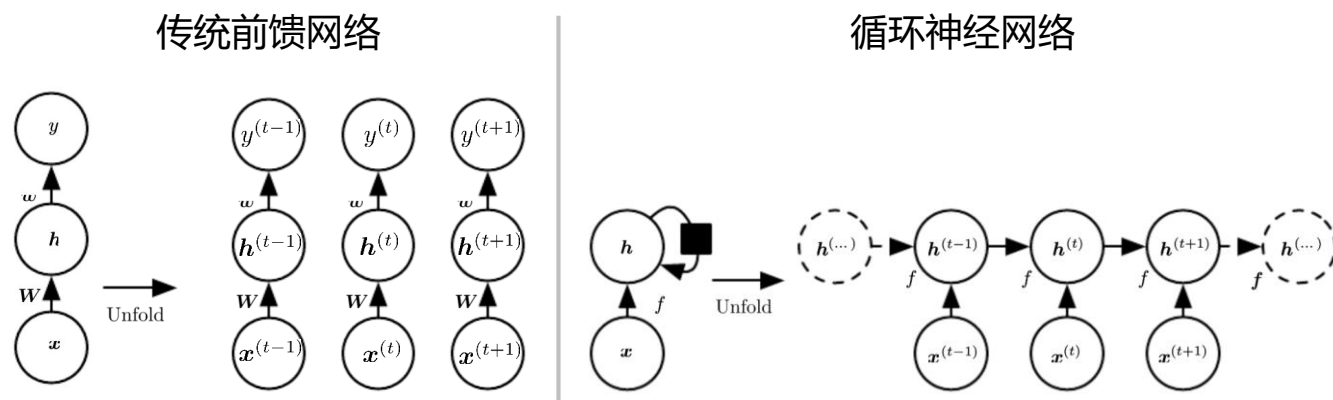


传统前馈网络用来处理序列数据

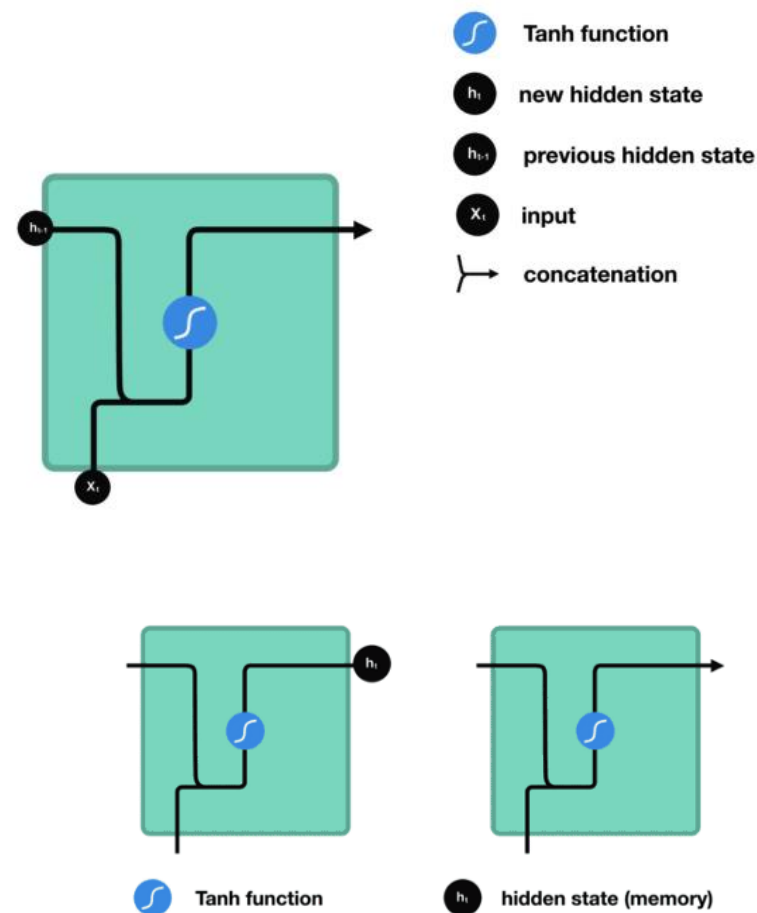
循环神经网络 (RNN) : 思想

思想:

- 在处理当前输入时, 把前一个输入的信息利用进来

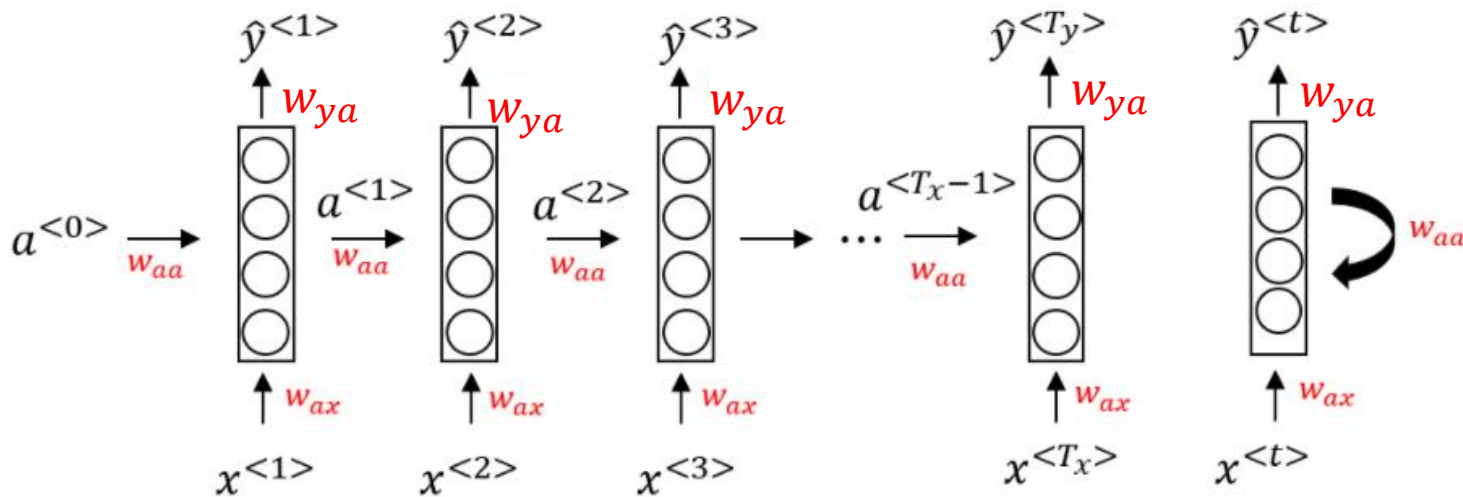


传统前馈网络 (左) 与循环神经网络 (右) 的比较



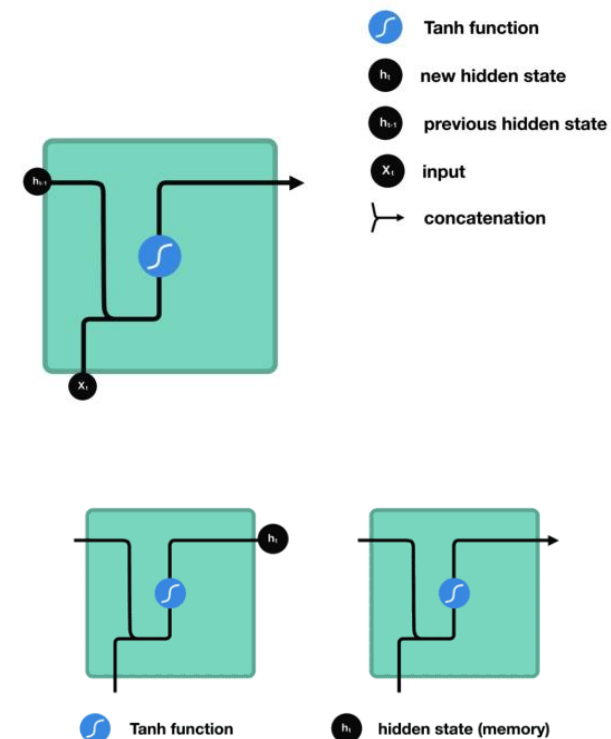
循环神经网络的前馈过程

循环神经网络 (RNN) : 训练



循环神经网络前馈过程详解

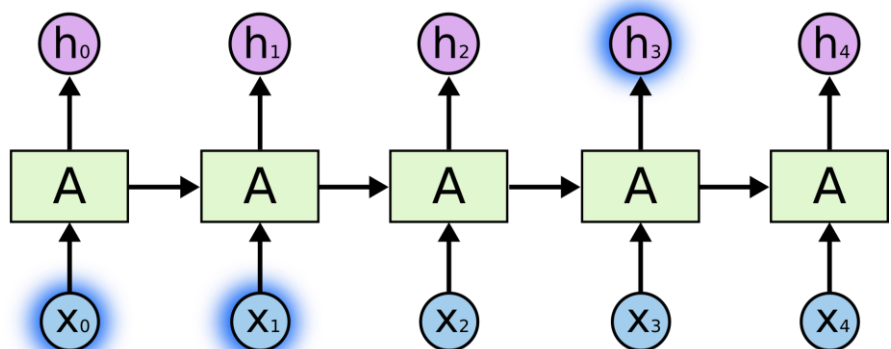
训练：通过反向传播，把权重矩阵 w_{ax} 、 w_{aa} 和 w_{ya} 训练出来即可



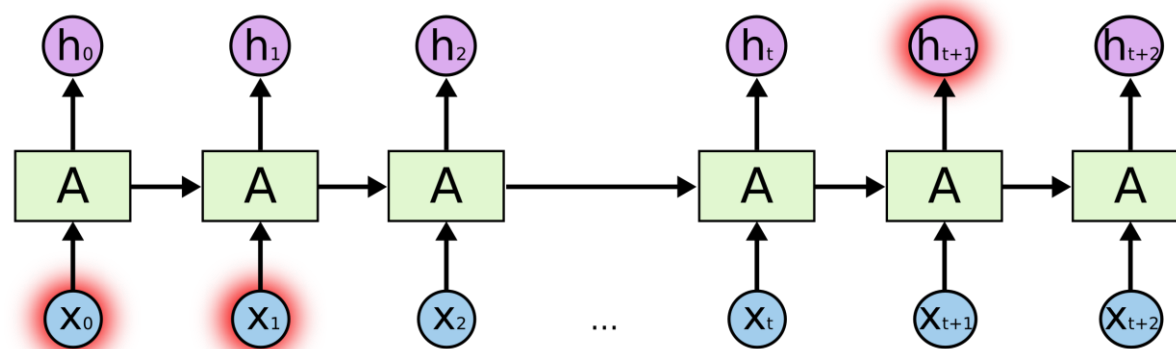
循环神经网络的前馈过程

循环神经网络 (RNN) : 不足

不足： 由于梯度消失和梯度爆炸，无法学习到长期的、时间跨度较大的信息



短期：尚可

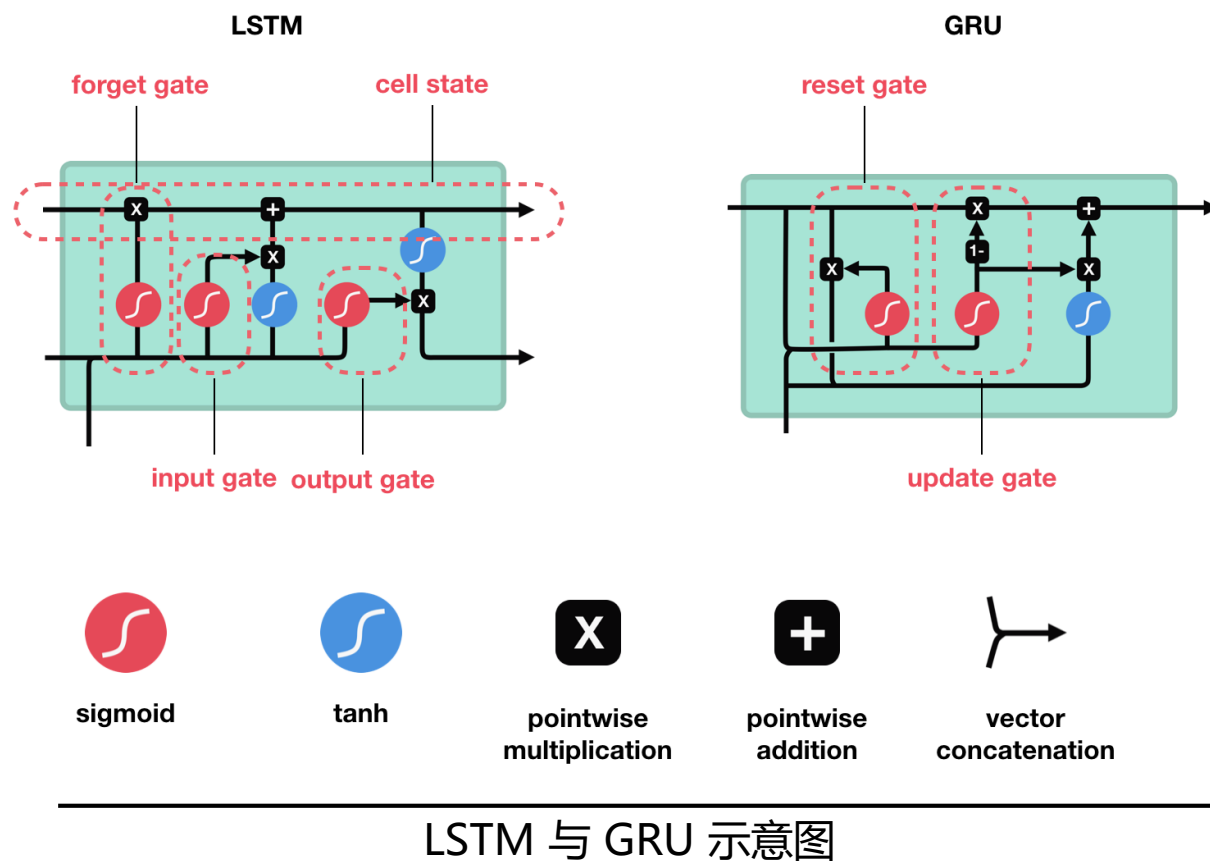


长期：学不到

长短时记忆 (LSTM) 与 循环门控单元 (GRU) : 思想

思想:

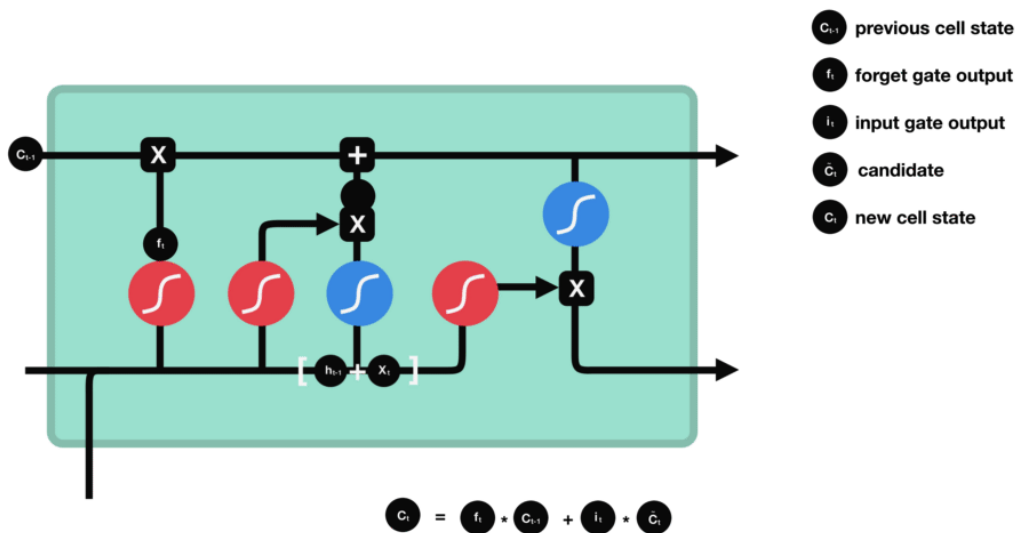
- 通过设计单元内的结构, 使得
 - 既能传递依赖关系
 - 又能改善梯度消失等问题
- LSTM (Long Short-Term Memory):**
新引入 **cell state** 的概念, 借助遗忘门和输入门来进行信息的跨时间传递
- GRU (Gated Recurrent Units):**
可视为LSTM的精简版本



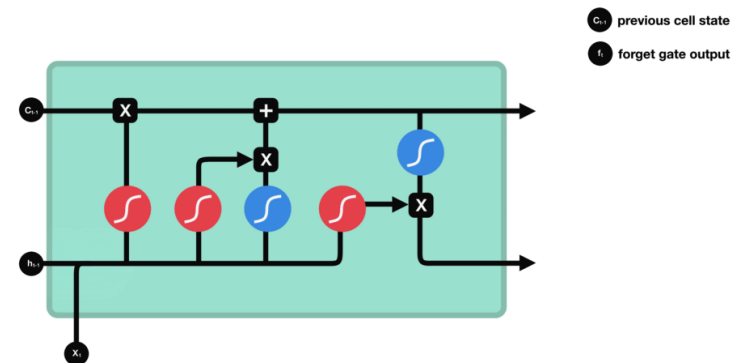


长短时记忆 (LSTM) : 简介

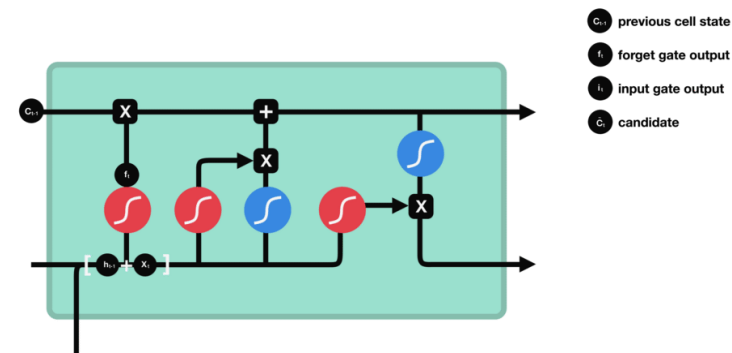
关键: cell state 的计算



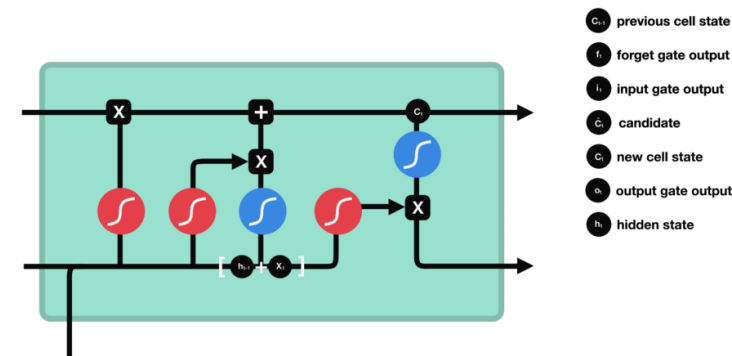
遗忘门
(forget gate)



输入门
(input gate)



输出门
(output gate)



长短时记忆 (LSTM) : 优势与不足

优势:

- 利用 cell state 机制, 能够学习到长期的依赖关系

不足: RNN的一些不足通常也存在于LSTM上, 如

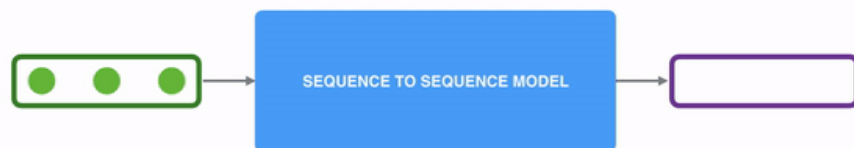
- 无法并行化: 必须一个输入接着一个输入地进行处理
- 当序列过长, LSTM 也不会表现得很好
 - 随着距离越远, 保持住上下文信息的概率呈指数下降

接下来:

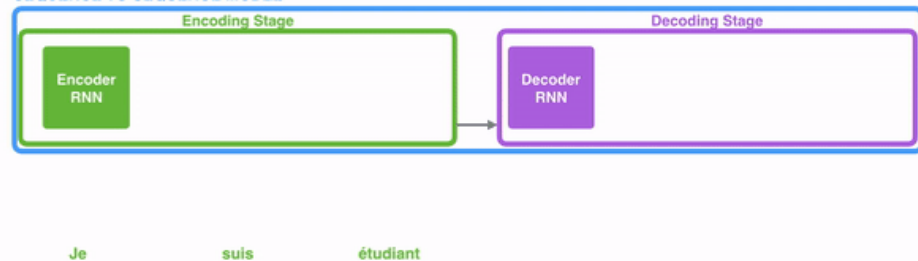
Transformer: 尝试结合 卷积神经网络 和 Self-attention机制 改善并行化的问题

讲在Transformer之前: Seq2Seq 与 Attention机制

Seq2Seq: 由 Encoder 和 Decoder 组成



Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL

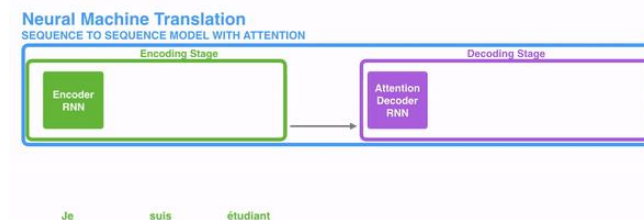


不足: Decoder只会利用到Encoder最后输出的那一个hidden state

Attention机制

思想:

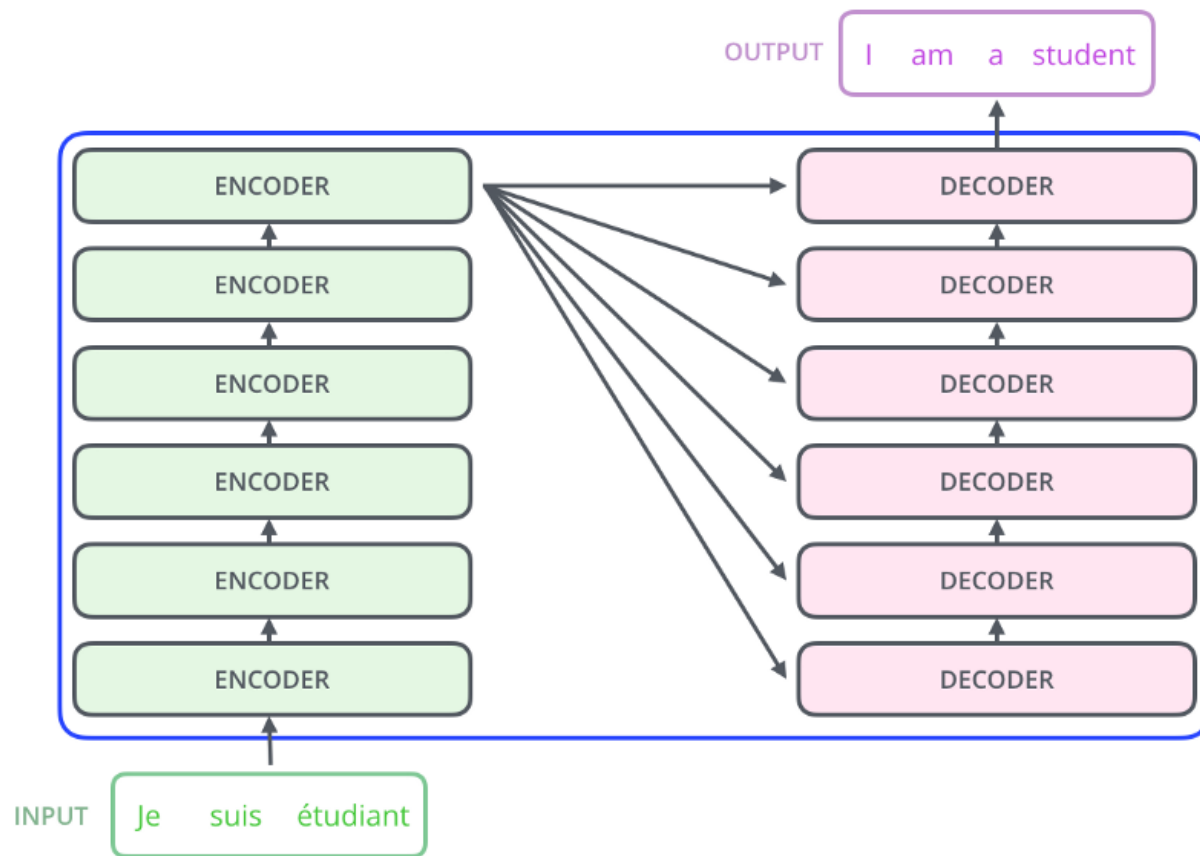
- 把 Encoder 中每一输入的hidden都利用上



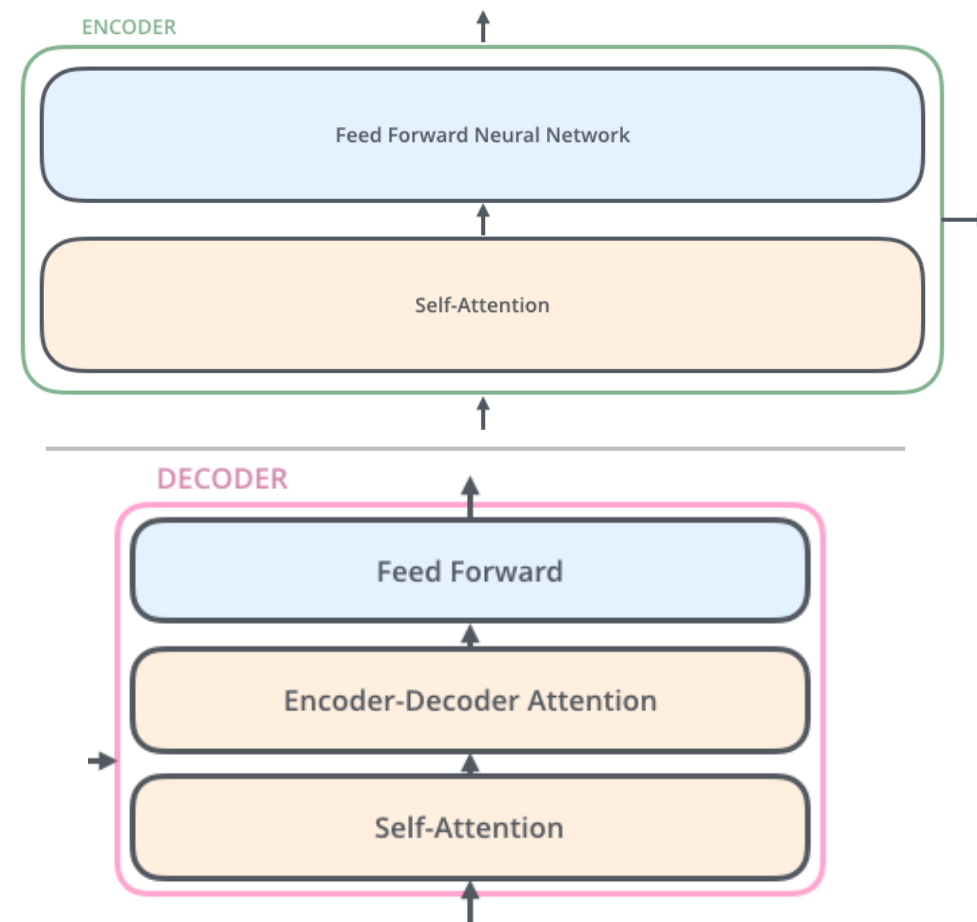
- 从而得以构建在Decoder的各个时间点处对各个输入的不同关注程度



Transformer: 本质是带有 Self-Attention机制的 Seq2Seq模型



Transformer 的基本结构



Encoder (上) 和 Decoder (下) 的内部结构

Transformer: Self-Attention

思想:

- 处理某一输入时，允许考虑它对其他输入的不同**关注程度**

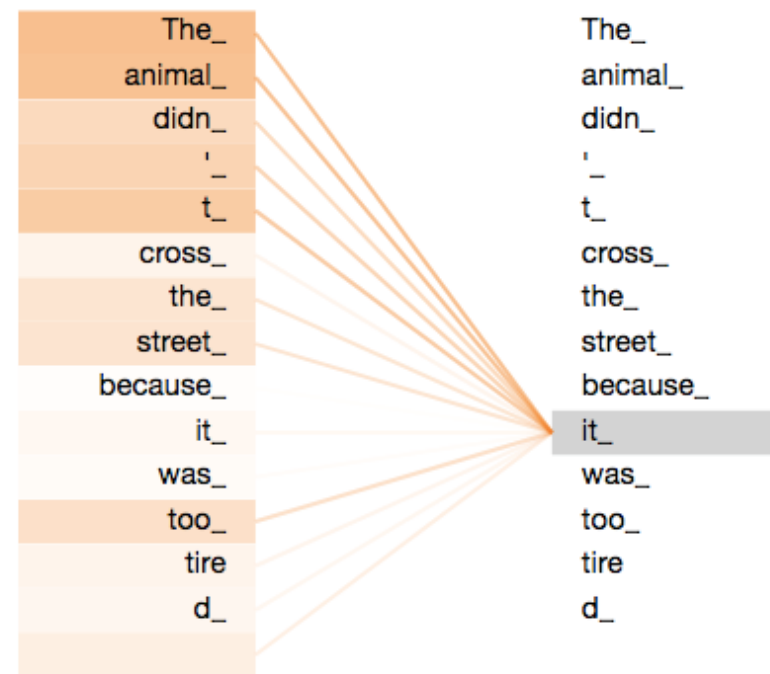
例:

- 以句子

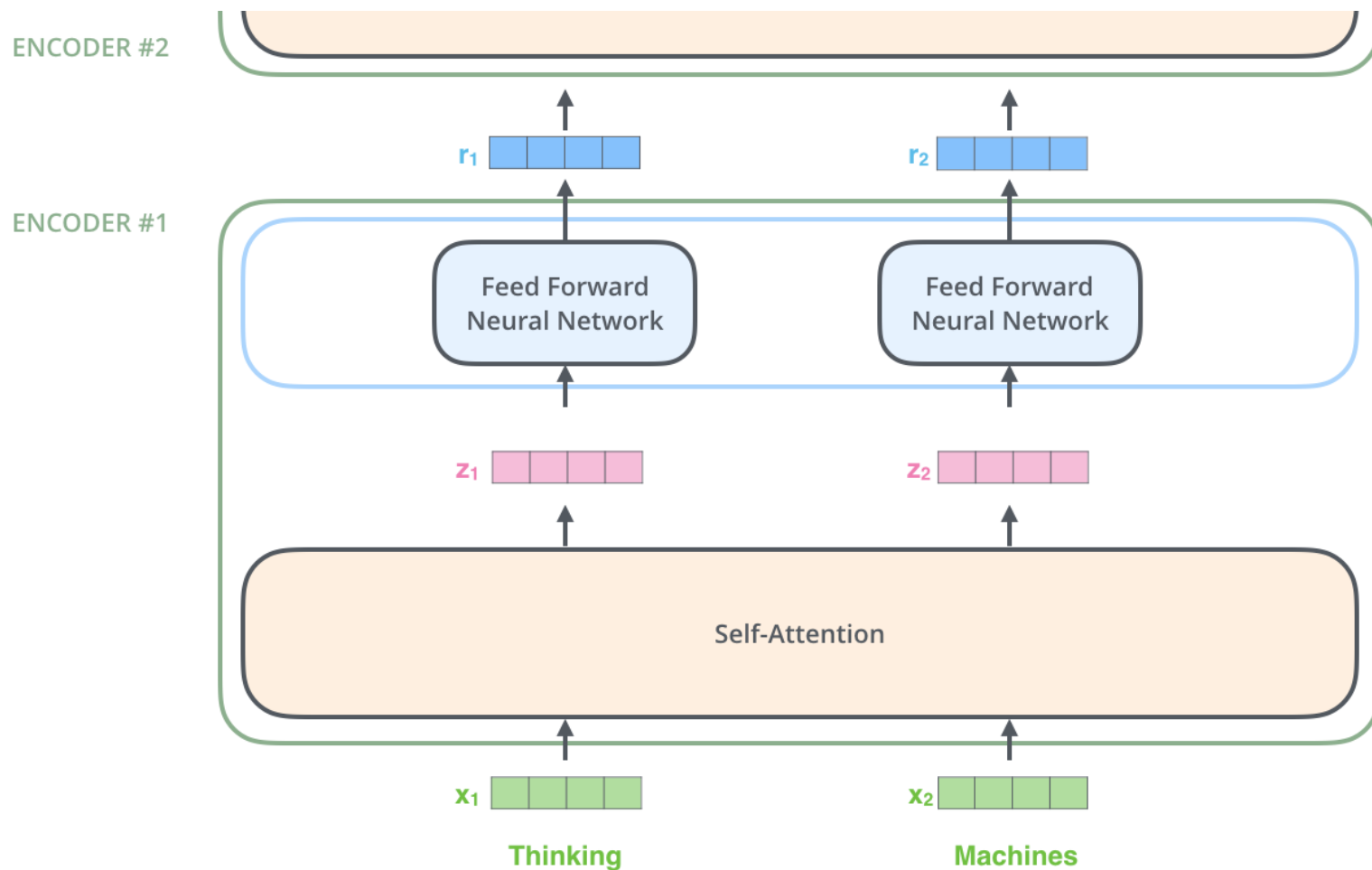
The animal didn't cross the street because it was too tired

为例，它是长度为 15 的**单词序列**（包含了句子结束符）

- 在处理输入 `it` 时，Self-Attention 机制允许模型通过训练权重矩阵从而挖掘出类似右图的**输入之间的关联程度**
- 从而能够更好地**利用其他词的信息**辅助对 `it` 进行编码

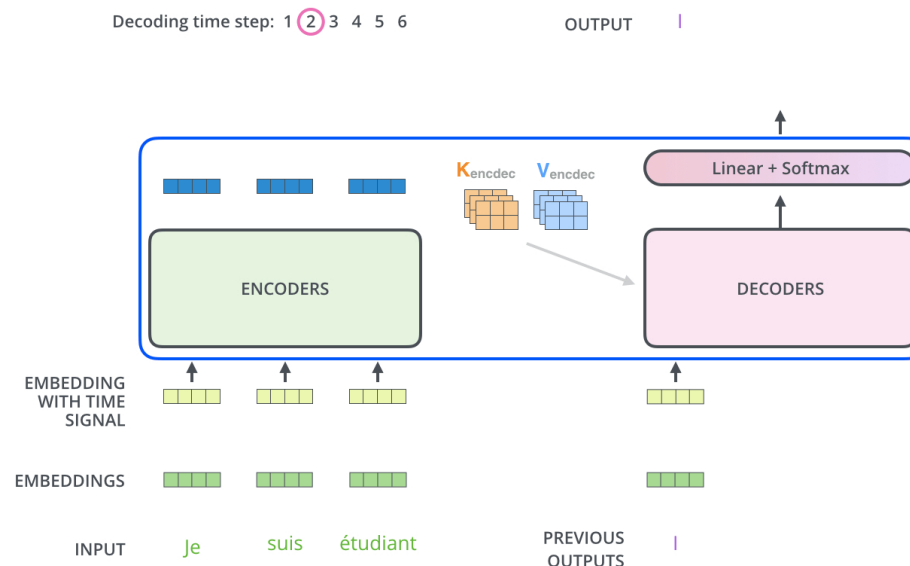
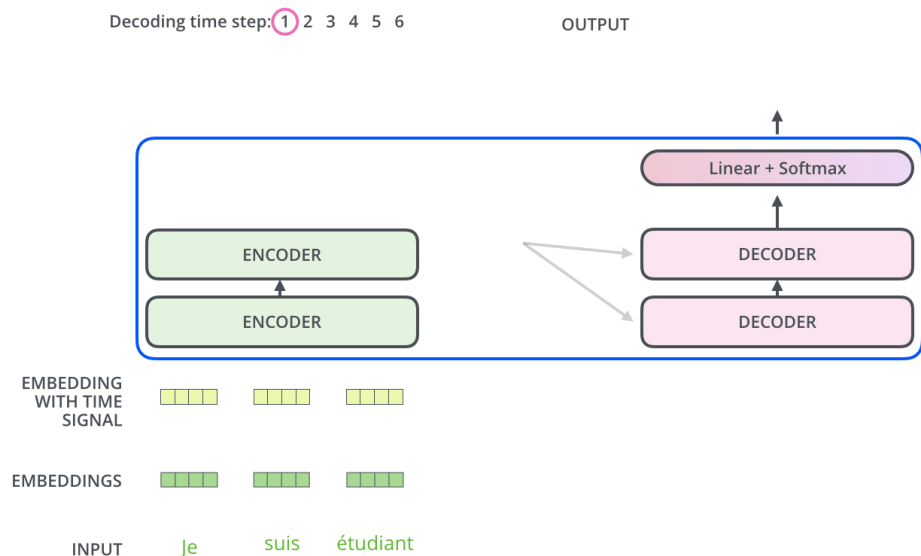


Transformer: Encoder部分所实现的功能



- 通过 Encoder 模块, 得到了两个向量 r_1 , r_2
- 也分别代表 Thinking、Machines 单词的信息, 但其为加权后的结果
- r_1 不仅仅包含 Thinking 单词的信息, 而且还有 Machines 单词的信息

Transformer: Decoder部分



输入序列经过 Encoder 部分，
最上面的 Encoder 的输出变换成一组 Attention 向量K, V
(用于每个 Decoder 的 Encoder-decoder Attention 层)

每个时间点的输出都在下一个时间点时
喂入给最底部的 Decoder，
直到 Decoder 输出结束符，结束

特点：Encoder 部分可以并行计算，一次性全部 encoding 出来

但 Decoder 部分不能并行，而是需要像 RNN 一样一个一个地解出来

03 | 对比实验

建模方案：数据理解 and 处理

序列数据：用自然语言处理问题 (NLP) 来理解

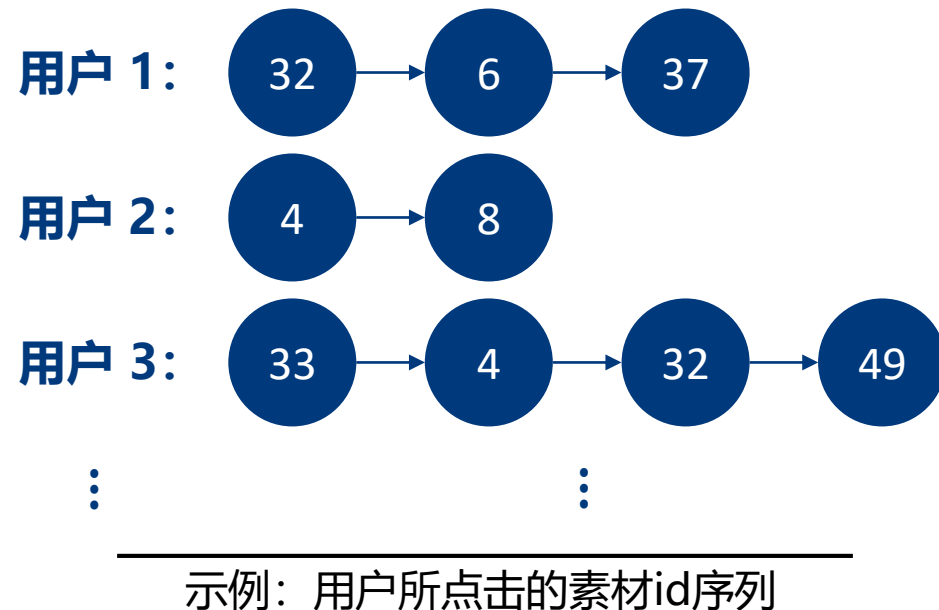
- 用户的点击序列可以理解成 NLP 中的句子
- 被点击的素材可以理解成句子中的词语

序列处理：

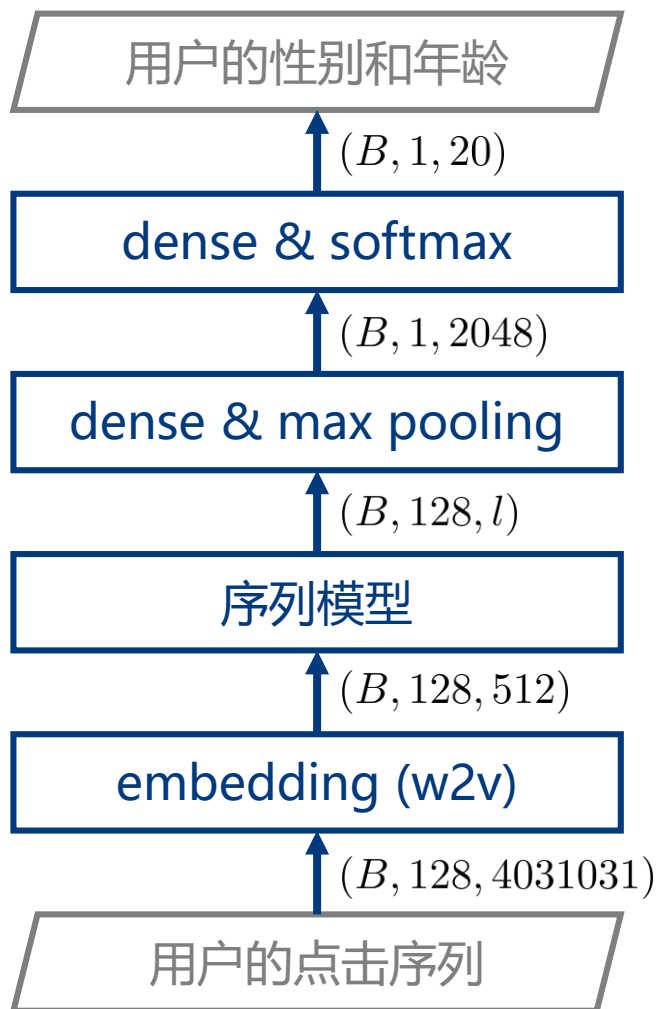
- 因为序列长度不一，点击序列只截取前 128 个
- 使用one-hot编码，每个用户的序列为 $128 \times d$ 的稀疏矩阵 ($d = 4,031,031$ 为素材id的总数目)

分类标签处理：

- 考虑成 20分类 问题
(年龄 10分类 \times 性别 2分类)



建模方案：整体思路

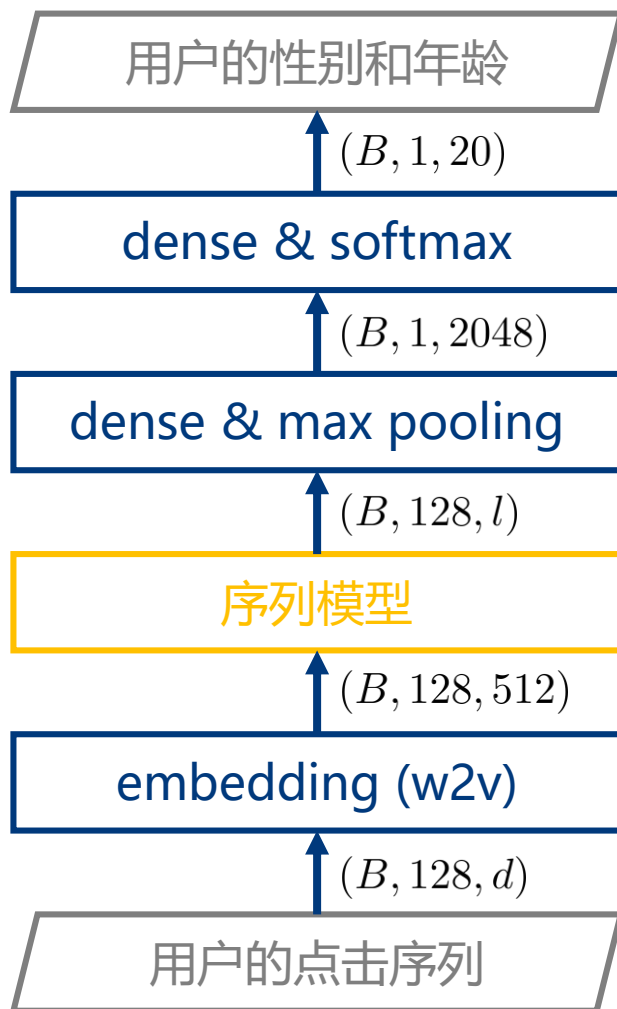


1. 进行word2vec的训练，从而可以将素材id由 d 维的 one-hot 编码转为 512 维的词向量表示
2. 将用户的长度为 128 的词向量序列作为模型输入，利用LSTM或Transformer等，建立 20分类模型
3. 预测时对20分类的概率进行聚合得到最终预测结果

$$P(\text{gender} = 1) = \sum_{i=1}^{10} P(\text{gender} = 1, \text{age} = i)$$

$$P(\text{age} = 1) = \sum_{j=1}^2 P(\text{gender} = j, \text{age} = 1)$$

建模方案：序列模型



将 不使用序列模型 和 使用非序列网络 作为对比基准:

- No Sequence Model (null)
- Deep Neural Networks (dnn)

进行实验的序列模型如下:

- GRU (gru)
- BiLSTM (lstm1)
- BiLSTM + BiLSTM (lstm2)
- RNN (rnn)
- Transformer (tr1)
- Transformer + Transformer (tr2)
- Transformer + BiLSTM (trlstm)

实验结果

	训练时长 (s)	预测时长 (s)	epoch	测试准确率之和	训练参数个数
null	1801.55	193.2019	4	1.289235	278548
dnn	5498.193	337.5896	4	1.433978	9489940
gru	8764.004	349.1615	4	1.428535	13690388
lstm1	9366.354	354.5557	4	1.439718	14741012
lstm2	11834.74	448.7291	4	1.436695	21040660
rnn	8112.528	318.7069	4	1.399139	11589140
tr1	9270.316	511.6164	4	1.438611	12642324
tr2	15918.23	634.1215	5	1.437045	15794708
trlstm	16103.33	606.9895	5	1.436599	17893396

实验初步结论

1. 使用 RNN 的表现甚至不如直接使用 DNN

这可能说明：如果序列依赖信息学得不好，甚至会导致最后模型效果更差

2. 使用 BiLSTM 和 Transformer 的模型表现都不错

一定程度上说明：这两个模型在序列建模上是很有竞争力的选项

3. GRU 的表现稍差

作为LSTM的精简版，在依赖信息的学习上可能还是略逊一筹，但胜在训练时间相对更短

下一步工作

1. 进行多次实验，取平均值作为最终的结果

由于时间关系，目前只做了一次实验；为了结果的可靠性，后面将再进行多次实验

2. 进行进一步的调参

Transformer 模型还有调参的空间，可能本次实验结果未能很好代表其实力

An aerial photograph of a coastal city, likely Xiamen, China. The city features a mix of modern and traditional architecture, with a prominent white building with a red roof in the center. The city is surrounded by greenery and a large body of water, with mountains visible in the background under a hazy sky.

Thank you

2020.12.07