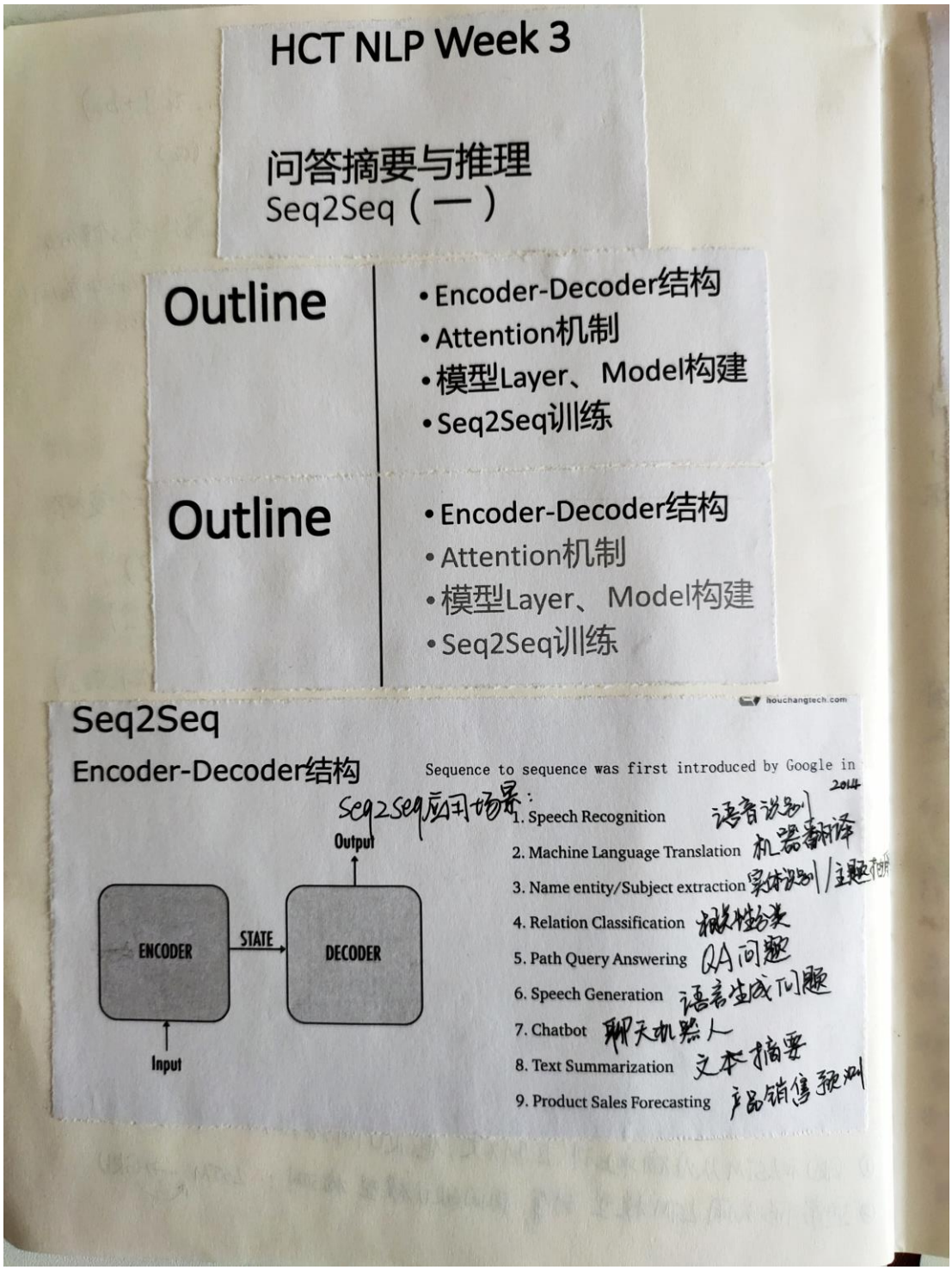
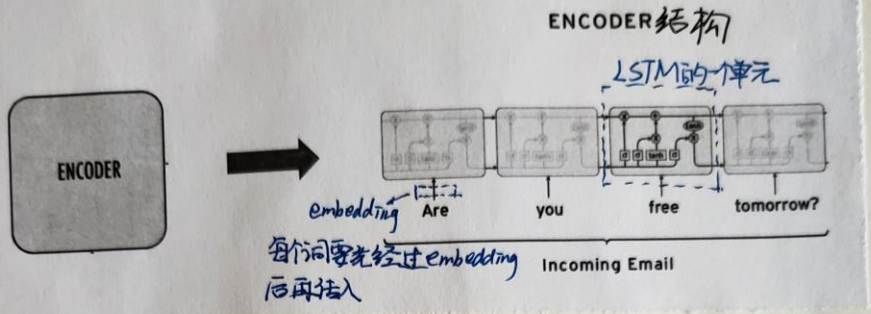
第三章 问答摘要与推理-Seq2Seq（一）



Seq2Seq

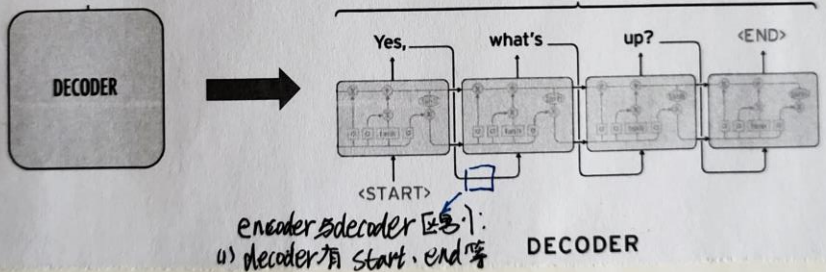
Encoder-Decoder结构



每个词要先经过embedding后再输入

Seq2Seq

Encoder-Decoder结构



encoder与decoder区别：

(1) decoder有start、end等特殊符号

(2) decoder会将前个词的输出作为后一个词的输入的一部分

houchangjieci.com

Seq2Seq

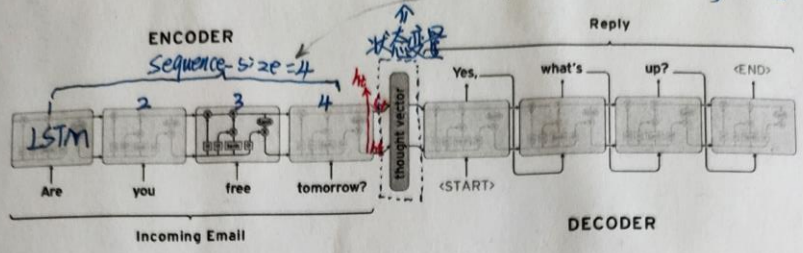
Encoder-Decoder结构

一次喂给神经网络的样本的个数

每个样本中词(词组)的个数

假设为300维

[batch_size, sequence_size, embedding_dim]

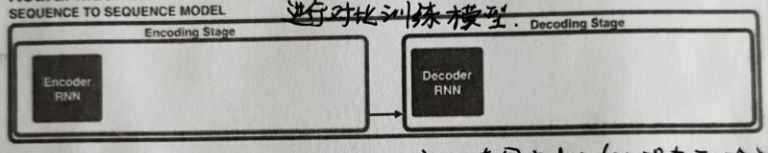


Seq2Seq

Encoder-Decoder结构

Seq2Seq问题：训练的时候有标签，预测的时候没有标签（用特殊符号 START、END表示起止）

Encoder训练过程是有监督的，训练结果为真实结果进行对比训练模型。



Encoding时传递的为状态变量

注：动态图中decoder没有画词输入

Je suis étudiant

Encoder-Decoder结构动态图网址

(source: Jay. Alammar, 2018)

Seq2Seq

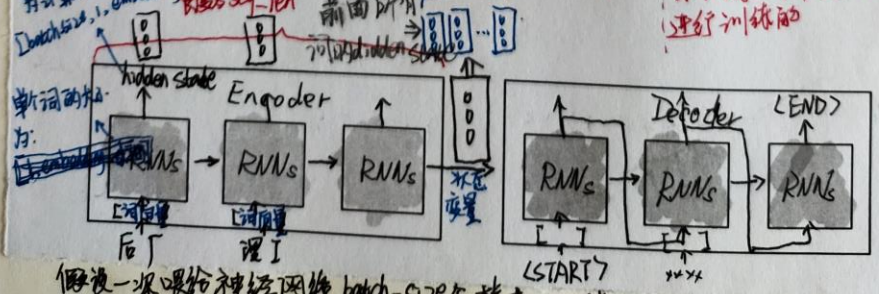
Encoder-Decoder结构

状态变量大小 [batch_size, seq_len, embedding_dim]

即状态变量，长度为seq_len

注：训练时，不是一个样本一个样本训练的，而是一个batch_size个样本训练的

前面所有词的hidden state



假设一次喂给神经网络 batch_size个样本，每个样本中词的数量为 seq_len，则一次Encoder的维度为[batch_size, seq_len, embedding_dim]

注：① 假设有词表：今天/天气/很好/……，词表中总共有1万个词，记 $L_{voc}=1$万

② 设有样本：[今天，天气]
[今天，很好]
[天气，今天]
[天气，很好]

每个样本中词的数量为2，记 seq-len=2

③ 设样本总数为 L_{samp}，则 $L_{samp} \neq L_{voc}$，

④ 设 $L_{samp}=2$万，取 batch-size = 64，（即训练时每次喂给神经网络的样本数为64）

⑤ 设词向量维度为300维，则记 embedding_dim = 300，则

结论：① embedding 大小为：[1万, 300]

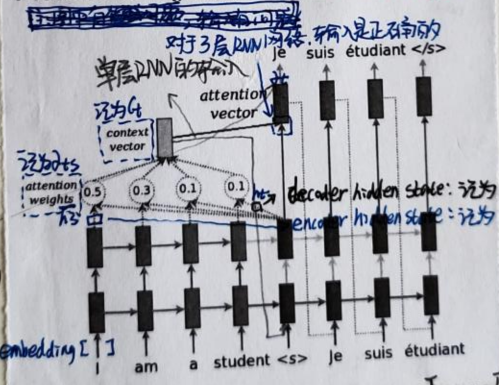
② Encoder-Decoder中，一次训练喂Encoder的大小为 [batch-size, seq-len, embedding-dim]

③ 中间的状态变量大小为 [batch-size, seq-len, embedding-dim]

Outline

- Encoder-Decoder结构
- **Attention机制**
- 模型Layer、Model构建
- Seq2Seq训练

Attention机制（先看下一张图，再看此图）



$c_t=\sum_s \alpha_{ts}\cdot\bar{h}_s$

如 $=0.5\bar{h}_{s_1}+0.3\bar{h}_{s_2}+\cdots$

$a_t=f(c_t,h_t)=\tanh(W_c[c_t,h_t])$

decoder hidden state：记为 h_t，encoder hidden state：记为 $\bar{h}_s$ } ⇒ 得到 Score $\xrightarrow{softmax}$ α_{ts}

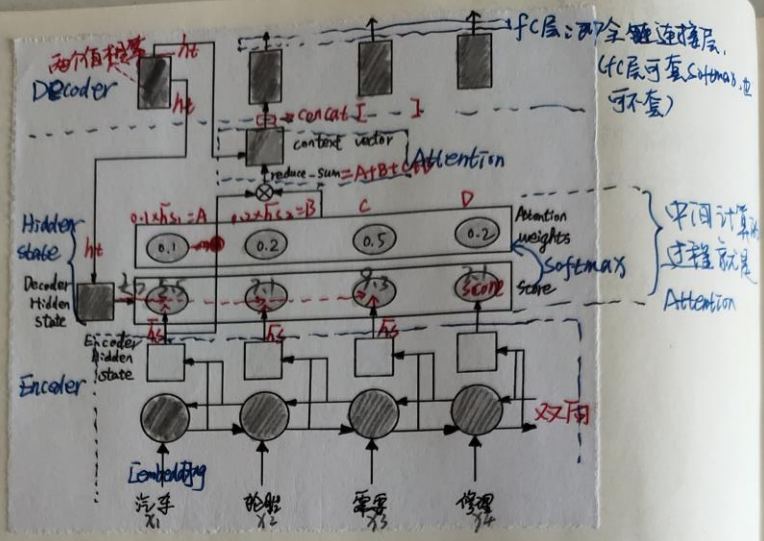
$$\alpha_{ts}=\frac{e^{score}}{\sum_{s=1} e^{score_s}}$$

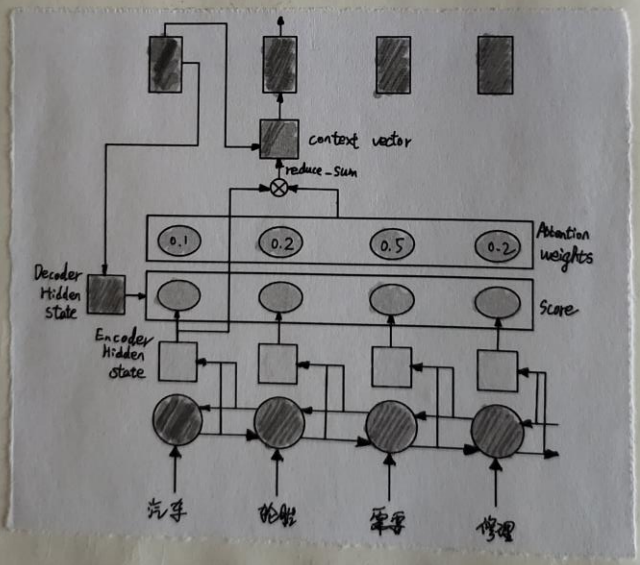
Score的计算方式有2种：

$$score=\begin{cases} h_t^T\cdot W\cdot\bar{h}_s & \text{(Luong's, multiplicate style)} \\ V_a^T\cdot\tanh\cdot(W_1 h_t+W_2\bar{h}_s) & \text{(Bahdanau's additive style)}\end{cases}$$

注：其中 W, W_1, W_2 为神经网络的权重矩阵，不是embedding

注：
① 摘要通常只做一层神经网络
② 每出一个词，context vector 都要重新计算
③ decoder 和 attention 是同时计算的
④ 在一次训练中，先encoder 再decoder





注：摘要抽取和翻译问题最大的不同：前者encoder、decoder词向量共享、后者不共享、前后者有两个数据集、两个词向量。

Outline

- Encoder-Decoder结构
- Attention机制
- **模型Layer、Model构建**
- Seq2Seq训练

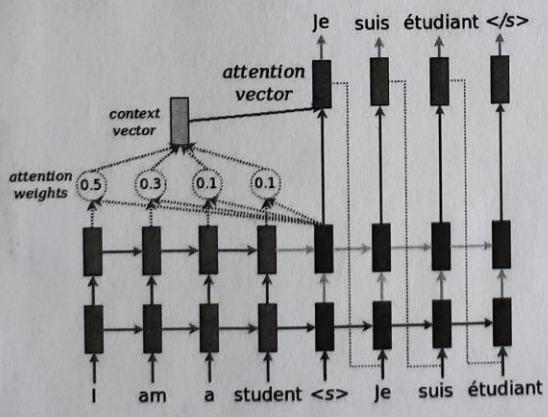
模型Layer、Model构建

代码见jupyter

Outline

- Encoder-Decoder结构
- Attention机制
- 模型Layer、Model构建
- **Seq2Seq训练**

Seq2Seq训练



How is it trained?

Seq2Seq训练

cross-entropy loss 交叉熵

交叉熵描述了两个概率分布之间的距离

注：神经网络的输出不一定是概率分布。往往使用softmax将输出变为概率分布

假设一次Decoder ~~输入了3个词~~ 输出为 $z=[y_1, y_2, y_3]$，

假设：z_1 输出为 3
$z_2 \longrightarrow 1$
$z_3 \longrightarrow -3$

$\xrightarrow{\text{Softmax}} \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}} = 0.88$, 0.12, 0.00 } Softmax输出的概率分布

〈概率一般不会为0〉 ~~解释~~ 即使概率为0，也不会出现计算错误

S(y) 预测值.

0.88
0.12
0.00

$$D(S,L) = -\sum_i L_i \log(S_i)$$

交叉熵

L 对应的真实标签

1
0
0

随着Decoder序列长度的增加，Loss函数取交叉熵均值.

$$J = -\frac{1}{N}\left(\sum_i^N y_i \log \hat{y}_i\right)$$

其中 $\hat{y}_i$ 为预测值，y_i 为真实标签，N为decoder序列的长度

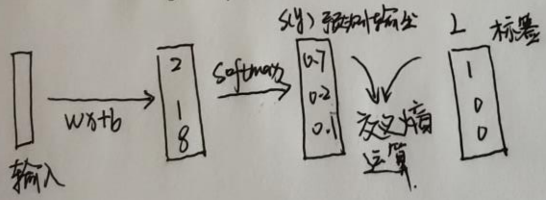
即：
$$L = -\frac{1}{N}\left(\sum_i^N y_i \log \hat{y}_i\right)$$

对于二分类问题：

$$L = -\frac{1}{N}\sum_1^N \left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]$$

y的取值为0或者1.

整个训练过程可如下表示：



Seq2Seq训练

解决摘要问题中生成的词没在文中出现过，句子很乱，不通顺的问题

其他训练技巧：(1)先验知识、 → 在decoder过程中加入先验知识，使得选取词为文中出现过的词

decoder $y \Rightarrow softmax(y) \Rightarrow P_i = \frac{e^{y_i}}{\sum_j e^{y_j}}$

先验知识引入：设字典大小为 $|V|$，为 0/1 向量表示先验知识，构建数量

$x = (x_1, x_2, \dots, x_n)$ （先验向量）

$x_i = 1$：表示在文中出现过

$x_i = 0$ 表示没在文中出现过

在decoder时通过平移(shift)引入先验知识：

$\hat{y} = s \otimes x + t = (s_1x_1+t_1, s_2x_2+t_2, \dots, s_nx_n+t_n)$

学习参数（训练参数），要通过神经网络进行训练

$\frac{y+\hat{y}}{2} = y \Rightarrow softmax \to P_i = \frac{e^{y_i}}{\sum e^{y_i}}$

（y：最终输出）

注：引入先验知识即使取出的词更多的都来自源文中

Seq2Seq训练 技巧

(2)Teacher Forcing 即将前一步输出的真实结果作为下一步的输入

~~RNN模型是用前一步的输出作为输入~~

Teacher Forcing方法能够解决收敛速度慢和不稳定的问题

Extensions to Teacher Forcing

普通训练：

<START> 奔驰 汽车 的 方向盘 <END>

<START> 预测成了的 <START>的 ?

第二个输出本来应该为"汽车"，错误预测成了 的，再将"<START>的"传入进行预测，导致错的越来越多（不收敛）

Teacher forcing:

<START> 奔驰　汽车　的　……　<END>

↑ <START>　↑ <START>奔驰　↑ <START>奔驰,汽车.

Teacher forcing 即不管预测是否对错,都将上一步的正确值代入下一步的输入.

注:预测时采用teacher forcing会带来exposure bias问题(即如果前面错了,后面会跟着错。

Seq2Seq训练

Exposure Bias ↘ 在训练过程中产生的

预测序列概率:

$P(y|x) = P(y_1|x)P(y_2|x,y_1)P(y_3|x,y_1,y_2)\cdots P(y_n|x,y_1,\dots,y_{n-1})$

解决ExposureBias:

~~训练过程中存在~~

训练 ⟶ inference(推理)

(1) 在推理过程中(测试)解决,,即传入2个或多个概率最大的代入下一步

(2) ① Scheduled Sampling (前面几步给真实值、后面decoder步骤给真实值、给预测值)

② RL:强化学习(Reinforcement Learning)

③ 构建负样本(类似Mask操作)

④ 对抗训练(梯度惩罚)

附：

（1）OpenNMT：Open source ecosystem for neural machine translation and neural sequence learning：https://github.com/OpenNMT

（2）Encoder、Decoder、Attention 层解说博文：
https://blog.csdn.net/zimiao552147572/article/details/105893842

（3）TensorFlow 官方 attention 文档“基于注意力的神经机器翻译”：
https://www.tensorflow.org/tutorials/text/nmt_with_attention

（4）Google Cloab：
https://colab.research.google.com/github/tensorflow/docs-l10n/blob/master/site/zh-cn/tutorials/text/nmt_with_attention.ipynb

注：从如下接口进入 Google Cloab