NLP-BERT

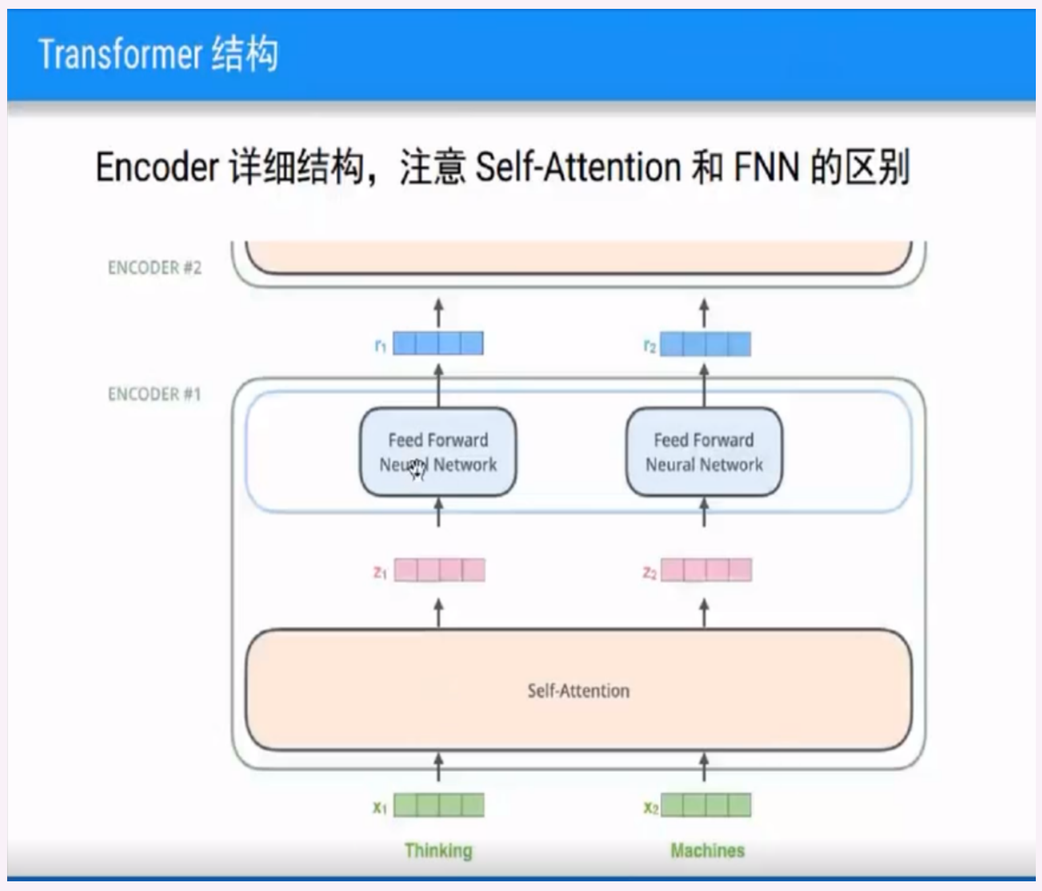
2019年7月2日　　20:55

传统RNN词向量：1.稀疏矩阵：区分性不大，猪和石头的区分度和猪和牛的区分度是一样的[0,1,0,0,0]这样的
2.稠密矩阵，但一次性只能保持3-5个单词的记忆，太多会使参数过多

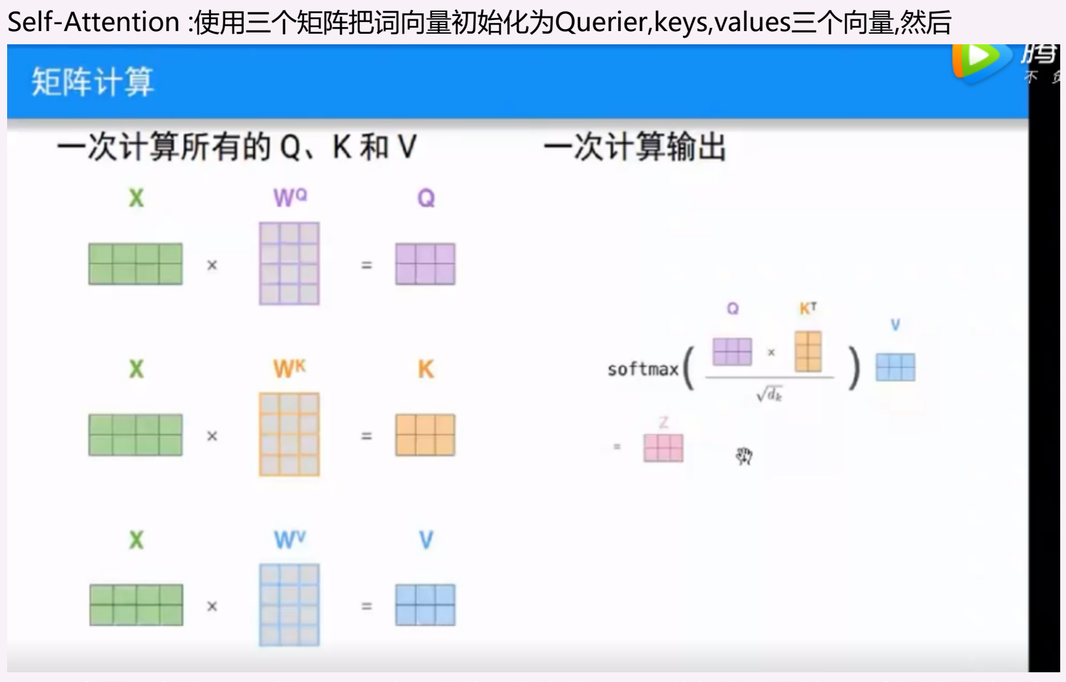
词向量学习：根据多个上下文的情况，如我要去北京/上海/天津，词向量是相似的

Word2vec：

1. 两个词上下文相似，则语义相似
2. CBOW:根据上下文推测中间词　Skip-Gram:根据中间词推测上下文
3. 解决多义词问题使用RNN，考虑t-1时刻的状态才得出t时刻的状态，具有记忆能力
4. 由于梯度消失 普通的RNN很难实现长距离的记忆，于是有了LSTM 。通过门，把一些重要词汇存在门中留在内存中，直到遇到需要发挥作用的时间取出，得到推测结果后，可以删掉这个词（理想情况）。GRU是把遗忘门和输出门合并成一个们简化模型
5. 输入输出长度不同可以使用两个RNN，enocder & decoder。Seq2seq如机器翻译
6. RNN问题：顺序依赖，无法并行 & 单向信息流
7. Attention普通的需要外部驱动，
8. Transformer：可以看到z1计算是需要x1,x2,x3… z2相同，但是计算r1,r2时就可以只是用z1,z2



9. Self-Attention :使用三个矩阵把词向量初始化为Querier,keys,values三个向量,然后



10. Elmo:多层双向的LSTM的NNLM,上下文编码作为特征，无监督学习语料不一定适合具体任务。
11. openAI GPT：根据特定任务改变编码，使用transformer替代RNN
12. BERT:词编码+位置编码+特定句子编码（是哪个句子中的）。Masked LM,遮住一些内容，让bert预测，让他能考虑到前后信息。50%抽取连续数据，50%抽取不连续句子，可学习两者关系
13. 定长序列的马尔可夫模型：（计算这个句子的概率，而非句子中每个词的概率）

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$
$$= P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1})$$
$$= P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})$$

即首先要知道第一个词的概率，然后一个一个推导，当然也可以选择两个词概率得到第三个词概率。

14. 变长序列的马尔可夫模型：把STOP也加入词库中，这样所有长度的句子概率都会有。生成第一个词是p(X1 = x1|X0=*)，长度为1的句子概率为1-p(STOP|*)
    a. 假设词库有10000个词，那么q(w|u,v)可能取值是10000*3，且使用最大似然估计，也即训练集的条件概率进行预测，则会有很多概率为0的序列，甚至分母为0，这是不合法的
15. 为解决15，可以将有概率较大的序列的值分给其他的序列。

    $c^*(v,w)=c(v,w)-\beta$

    其中β是一个0-1之间的数。我们用这个打折后的$c^*(v,w)$来计算$p(w|v)$：

    $p(w|v)=c^*(v,w)/c(v)$

    将其余概率分配给其他未出现的序列，一个比较好的思想是如果w'在训练集出现概率较大但（w' , v）未出现，则将概率分给(w',v)

    多余的概率为；

$$\alpha(v) = 1 - \sum_{w:c(v,w)>0} \frac{c^*(v,w)}{c(v)}$$

$$\mathcal{A}(v) = \{w | c(v,w) > 0\}$$
$$\mathcal{B}(v) = \{w | c(v,w) = 0\}$$

$$q_D(w|v) = \begin{cases} \frac{c^*(v,w)}{c(v)} & \text{If } w \in \mathcal{A}(v) \\ \alpha(v) \times \frac{q_{ML}(w)}{\sum_{w \in \mathcal{B}(v)} q_{ML}(w)} & \text{If } w \in \mathcal{B}(v) \end{cases}$$

16. 语言模型通常使用困惑度即perplexity来评价

$$PPL = 2^{-\frac{1}{M}\sum_{i=1}^{m} log_2 p(x_i)}$$

    m是句子个数,M是语料库单词个数，p(xi)是语言模型输出这个句子的概率

17. 所以Attention机制的目标**是帮助我们自动找出RNN的哪个时刻Cell的输出是强特，如果是RNN的输入是一个句子，我们就希望Attention机制能够帮我们找出，句子中的哪个词是比较关键的词** 。通俗的说法就是Attention机制使模型在做任务时，将注意力主要集中在了对任务有帮助的的重要的特征上面。