NLP->ATTENTION | 具有注意力机制的seq2seq模型

原创 VK 磐创AI 2020-05-27

收录于话题
#往期原创技术文

149个

磐创AI分享

来源 | Renu Khandelwal

编译 | VK

来源 | Towards Data Science

在本文中，你将了解：

- 为什么我们需要seq2seq模型的注意力机制？
- Bahdanua的注意力机制是如何运作的？
- Luong的注意力机制是如何运作的？
- 什么是局部和全局注意力？
- Bahdanua和Luong注意力机制的关键区别

什么是注意力，为什么我们需要seq2seq模型的注意力机制

让我们考虑两个场景，场景一，你正在阅读与当前新闻相关的文章。第二个场景是你正在阅读准备考试。两种情况下的注意力水平是相同还是不同？

与新闻文章相比，你在准备考试时会相当注意阅读。在准备测试的时候，你会更加关注关键词来帮助你记住一个简单或复杂的概念。这也意味着我们要专注于某一特定领域的任何深度学习任务。

序列到序列(Seq2Seq)模型使用编码器-解码器架构。

seq2seq的几个场景

- 神经机器翻译(NMT)
- 图像字幕
- 聊天机器人
- 文本摘要等

Seq2Seq模型将源序列映射到目标序列。在神经机器翻译的情况下，源序列可以是英语，目标序列可以是印地语。

我们将英语源语句传递给编码器；编码器将源序列的完整信息编码为单个实值向量，也称为上下文向量。然后，这个上下文向量被传递到解码器上，以生成目标语言(如印地语)中的输出序列。上下文向量负责将整个输入序列汇总为单个向量。

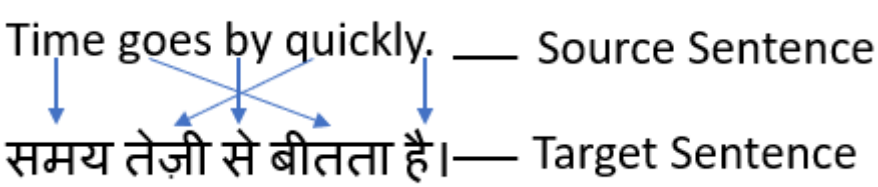
如果输入的句子很长，那么编码器中的一个向量可以保存所有要提供给解码器的相关信息吗?

在预测目标词时，是否可以将注意力集中在句子中的几个相关词上，而不是集中在包含整个句子信息的单个向量上?

注意力机制有助于解决问题。

注意力机制的基本思想是避免试图为每个句子学习单一的向量表示，而是根据注意力权值来关注输入序列的特定输入向量。

在每一解码步骤中，解码器将被告知需要使用一组注意力权重对每个输入单词给予多少“注意”。这些注意力权重为解码器翻译提供上下文信息

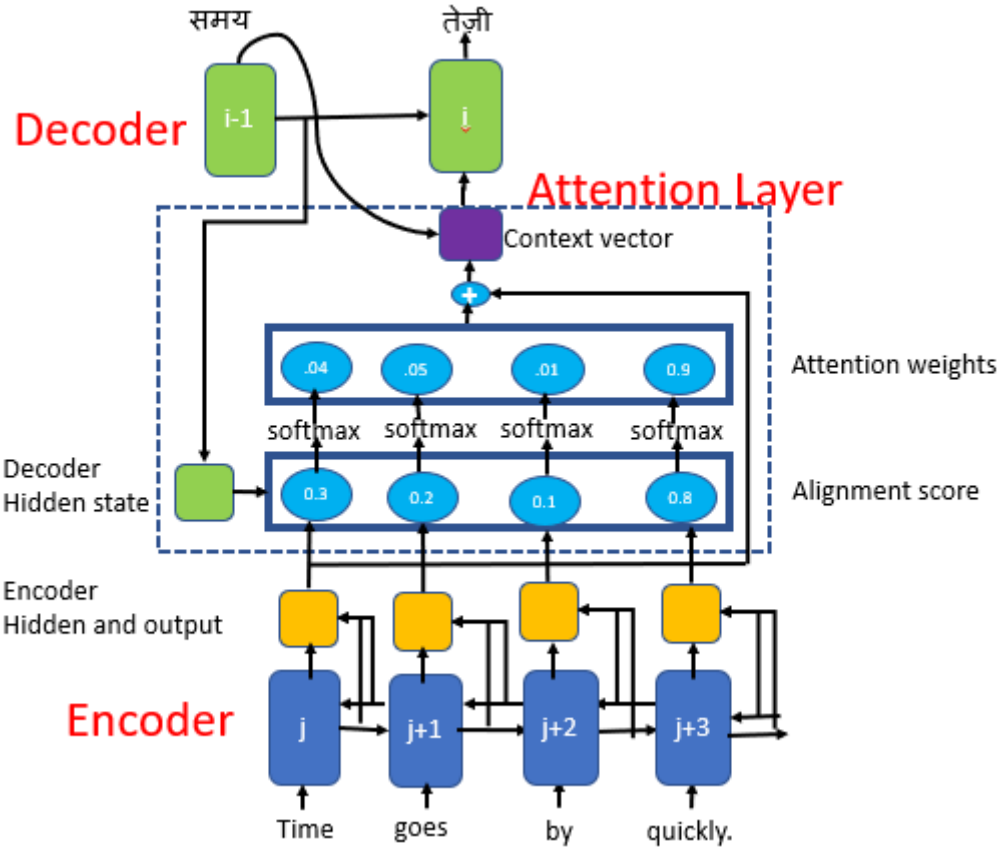


Bahdanau的注意力机制

Bahdanau等人。提出了一种学习结合对齐和翻译的注意力机制，它会执行编码器状态和解码器状态的线性组合。

让我们学习Bahdanau提出的注意力机制

- 编码器和解码器的所有隐状态(向前和向后)都用于生成上下文向量，这与seq2seq中仅使用最后一个编码器的隐状态不同。
- 注意力机制将输入和输出序列与前馈网络参数化的对齐。它有助于关注源序列中最相关的信息。
- 该模型根据与源位置相关联的上下文向量和先前生成的目标词来预测目标词。



带有注意力机制的Seq2Seq模型由编码器、解码器和注意力层组成。

注意力层包括

- 对齐层
- 注意力权重
- 上下文向量

对齐分数(Alignment score)

对齐分数映射位置“j”周围的输入与位置“i”处的输出匹配的程度。分数是基于前一个解码器的隐状态$s_{(i-1)}$，就在预测目标单词和输入句子的隐状态h_j之前

$$e_{ij.} = a(s_{i-1}, h_j) \qquad \text{Alignment Score}$$

解码器决定它需要关注源语句的哪个部分，而不是让编码器将源语句的所有信息编码成一个固定长度的向量。

对齐向量与源序列具有相同长度并在解码器的每个时间步被计算

注意力权重

我们将softmax激活函数应用于对齐分数，以获得注意力权重。

$$\alpha_{i_j} = \exp(e_{ij}) \Big/ \sum_{k=1}^{Tx} \exp(e_{ik}) \qquad \text{Attention weight}$$

Softmax激活函数将得到和等于1的概率，这将有助于表示每个输入序列的影响权重。输入序列的注意力权重越高，对目标词预测的影响越大。

在我们的例子中，我们看到一个更高的输入词的注意力权重值可以快速地预测目标词，तेज़ी

上下文向量

上下文向量用于计算解码器的最终输出。上下文向量c_i是注意力权重和编码器隐状态(h_1，h_2，...，h_{tx})的加权和，它映射到输入语句。

$$C_i = \sum_{j=1}^{Tx} \alpha_{ij} h_j \quad \text{Context vector}$$

预测目标词

为了预测目标词，解码器使用

- 上下文向量(c_i)，
- 上一时间步的解码器输出(y_{i-1})和
- 前一解码器的隐状态(s_{i-1})

$$s_i = f(s_{i-1}, c_i, y_{i-1})$$

Luong注意力机制

Luong的注意力模型也被称为乘法注意力。它通过简单的矩阵乘法将编码状态和解码状态降为注意力得分。简单的矩阵乘法使它更快，更节省空间。

根据注意力在源序列中的位置，Luong提出了两种类型的注意力机制

1. 全局注意力，关注所有来源位置
2. 局部注意力，注意力只放在每个目标词的源位置的一小部分上

全局注意力与地方关注的共性

- 在每个时间步t，在解码阶段，全局和局部注意的两种方法都首先以堆叠的LSTM顶层的隐状态h_t作为输入。
- 这两种方法的目标都是导出上下文向量，以获取相关的源端信息，帮助预测当前的目标词y
- 注意力向量被输入到下一个时间步中，以告知模型过去的决策。

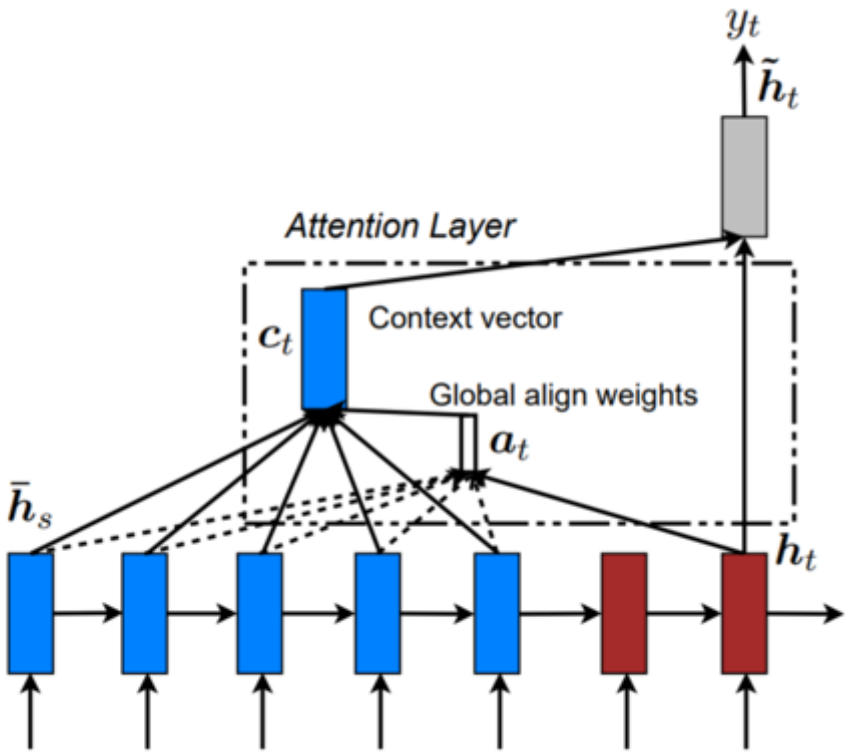
全局和局部注意模型的不同之处在于上下文向量是如何导出的

在讨论全局和局部注意之前，让我们先了解一下在给定的时间t内，Luong的注意力机制所使用的约定

- c_t：上下文向量
- a_t：对齐向量
- h_t：当前目标隐状态
- h_t：当前源隐状态
- y_t：预测当前目标字

- $h\tilde{}_{tt}$：注意向量

全局注意力



- 全局注意模型在计算上下文向量时考虑了编码器的所有隐状态。
- 通过比较当前目标隐状态h与每个源隐状态h的大小，导出了与源序列中的时间步数大小相等的可变长度对齐向量A
- 对齐分数被称为基于内容的函数，我们考虑三种不同的选择

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

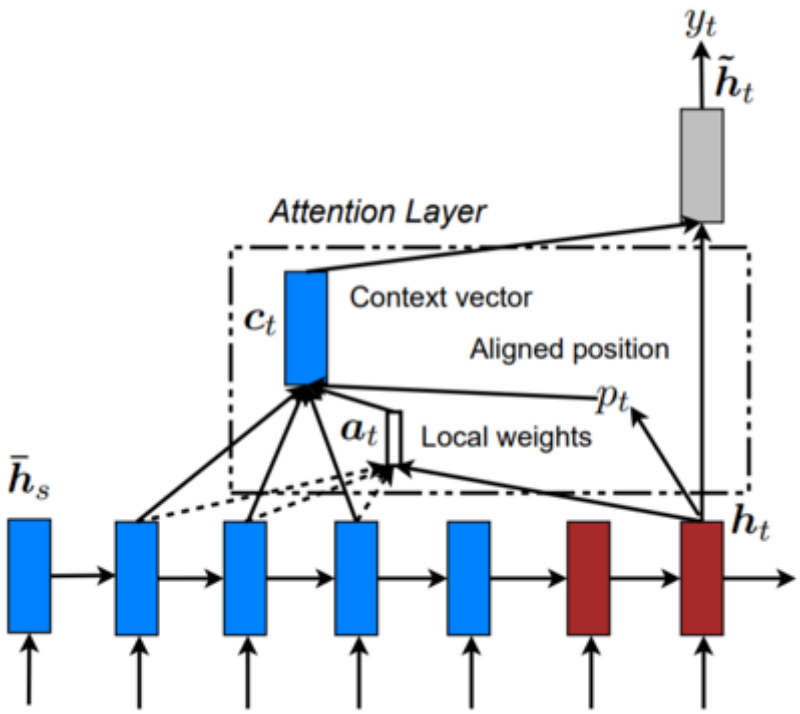
- 全局上下文向量c_t是根据所有源隐状态h_t上的对齐向量a_t计算的加权平均值

当源序列是一个大段落或一个大文档时会发生什么？

由于全局注意模型考虑了源序列中的所有单词来预测目标单词，因此在计算上变得非常昂贵，并且很难翻译出较长的句子

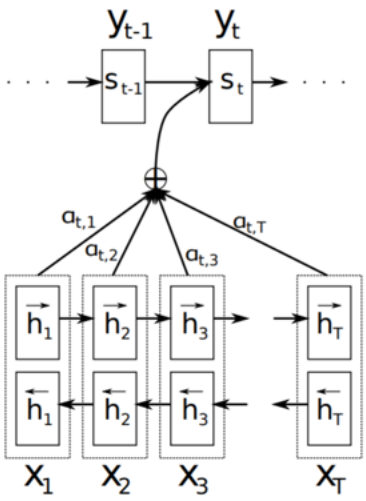
利用局部注意力可以解决全局注意模型的缺点

局部注意力

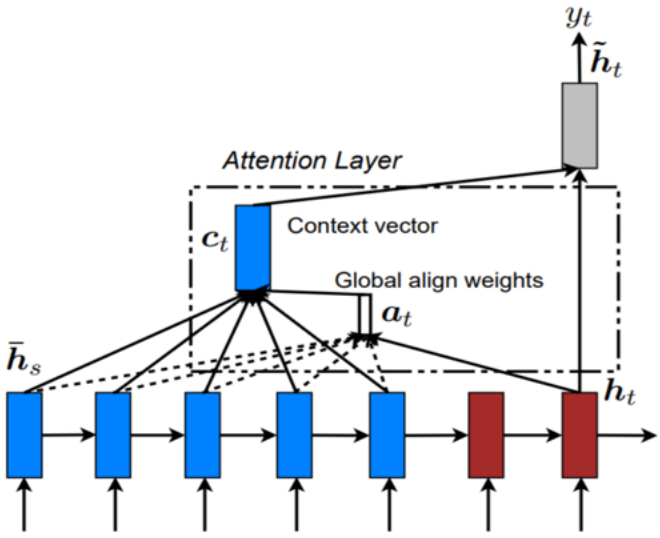


- 局部注意力只关注每个目标词的源位置的一小部分，不像全局注意力那样关注整个源序列
- 计算成本低于全局注意力
- 局部注意力模型首先在时间t为每个目标词生成对齐位置p_t。
- 上下文向量c_t是在选定窗口内源隐状态集上作为加权平均值导出的
- 对齐的位置可以单调地或预先地选择

Bahdanau和Luong注意力机制的关键区别



Bahdanau attention mechanism



Luong attention mechanism

Bahdanau和long注意力机制中的注意力计算

Bahdanau等人，在双向编码器中使用前向和后向隐状态的串联，在其非堆叠单向解码器中使用前一个目标的隐状态

Loung等人，注意力使用编码器和解码器的顶层LSTM层的隐状态

Luong注意力机制使用当前解码器的隐状态来计算对齐向量，而Bahdanau使用上一个时间步的输出

对齐函数

Bahdanau只使用concat分数对齐模型，而Luong使用dot、general和concat对齐分数模型

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

有了注意力机制的知识，你现在可以构建强大的深层NLP算法。

原文链接：https://towardsdatascience.com/sequence-2-sequence-model-with-attention-mechanism-9e9ca2a613a