Deep Learning Model for NLP Task - July 5'th, 2019

MyToc

- Attention
  - Attention 背景知识
  - Attention 模型
  - Attention 模型在自然语言处理和计算机视觉上的应用
- Transformer
  - Transformer 背景知识
  - Transformer 架构介绍
  - Transformer 模型在自然语言处理和计算机视觉上的应用
- Bert
  - Bert 背景知识
  - Bert 模型介绍
  - Bert 模型在自然语言处理上的应用
- XLNet
  - XLNet 背景知识
  - XLNet 模型介绍
  - XLNet 模型在自然语言处理上的应用
- Transformer XLNet
  - Transformer XLNet 背景知识
  - Transformer XLNet 模型介绍
  - Transformer XLNet 模型在自然语言处理上的应用

Attention 机制

Attention 背景知识

机器翻译解决的是输入是一串源语言表达的一句话，输出是目标语言相对应的话的问题，如将法语中的一段话翻译成合适的英语。

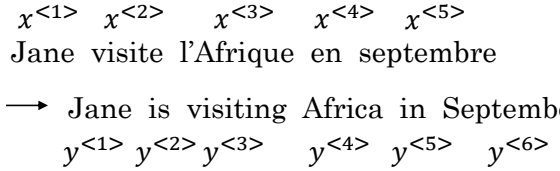

图 1：机器翻译例子（法语 ⟶ 英语）

之前的机器翻译 (简称 NMT) 模型中，通常的配置是 Encoder-Decoder 结构，即 Encoder 读取输入的句子并将其转换为定长的一个向量，然后 Decoder 再将这个向量翻译成对应的目标语言的文字。通常 encoder

及 decoder 均采用 RNN 结构，如 LSTM 或 GRU 等如下图所示，我们利用 encoder RNN 将输入语句信息总结到最后一个 hidden vector 中，并将其作为 decoder 初始的 hidden vector，利用 decoder 解码成对应的其他语言中的文字。

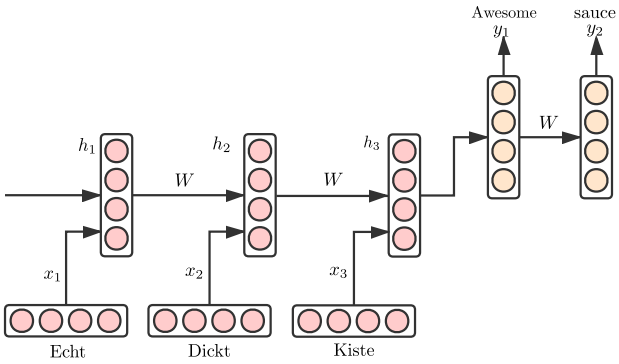

图 2：NMT 的 Encoder-Decoder 架构

但是这个结构有些问题，尤其是 RNN 机制实际中存在**长程梯度消失**的问题，对于较长的句子，我们很难寄希望于将输入的序列转化为定长的向量而保存所有的有效信息，所以随着所需翻译句子的长度的增加，这种结构的效果会显著下降。

以英语 - 法语翻译为例，给定一对英语输入序列“They”、“are”、“watching”、“.”和法语输出序列“Ils”、“regardent”、“.”。解码器可以在输出序列的时间步 1 使用更集中编码了“They”、“are”信息的背景变量来生成“Ils”，在时间步 2 使用更集中编码了“watching”信息的背景变量来生成“regardent”，在时间步 3 使用更集中编码了“.”信息的背景变量来生成“.”。这看上去就像是在解码器的每一时间步对输入序列中不同时间步编码的信息分配不同的注意力。这也是注意力机制的由来。它最早由 Bahanau 等提出。

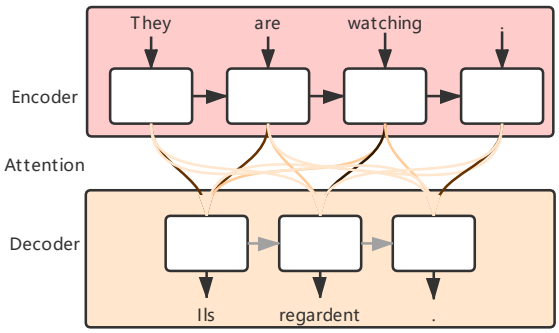

图 3：Attention + Encoder-Decoder 架构

上面是直观上的理解，下面来详细解释一下数学上有哪些具体的计算过程，和详细的算法流程。

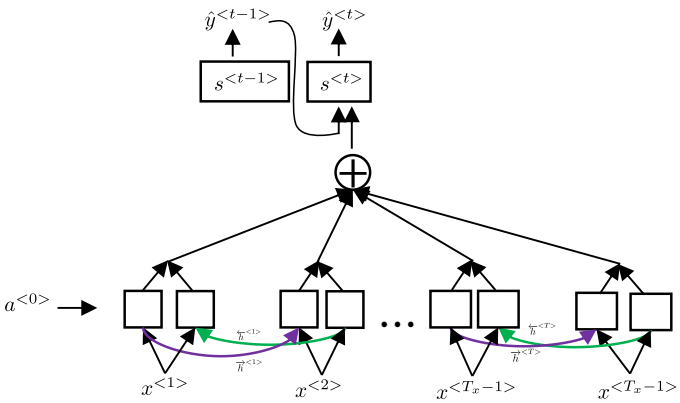

图 4：Attention 算子

- 首先我们利用 RNN 结构得到 encoder 中的 hidden state $(h_1, h_2, ..., h_T)$ ，

- 假设当前 decoder 的 hidden state 是 s_{t-1}，我们可以计算每一个输入位置 j 与当前输出位置的关联性，$e_{tj} = a(s_{t-1}, h_j)$，写成相应的向量形式即为 $\vec{e_t} = (a(s_{t-1}, h_1), ..., a(s_{t-1}, h_T))$，其中 a 是一种相关性的算符，例如常见的有点乘形式 $\vec{e_t} = \vec{s_{t-1}}^T \vec{h}$，加权点乘 $\vec{e_t} = \vec{s_{t-1}}^T W \vec{h}$，加和 $\vec{e_t} = \vec{v}^T tanh(W_1 \vec{h} + W_2 \vec{s_{t-1}})$ 等等。
- 对于 $\vec{e_t}$ 进行 softmax 操作将其 normalize 得到 attention 的分布，$\vec{\alpha_t} = softmax(\vec{e_t})$，展开形式为 $\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})}$
- 利用 $\vec{\alpha_t}$ 我们可以进行加权求和得到相应的 context vector$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})}$
- 由此，我们可以计算 decoder 的下一个 hidden state $\vec{\alpha_t}$ 以及该位置的输出 $\vec{c_t} = \sum_{j=1}^{T} \alpha_{tj} h_j$ 。

这里关键的操作是计算 encoder 与 decoder state 之间的关联性的权重，得到 Attention 分布，从而对于当前输出位置得到比较重要的输入位置的权重，在预测输出时相应的会占较大的比重。

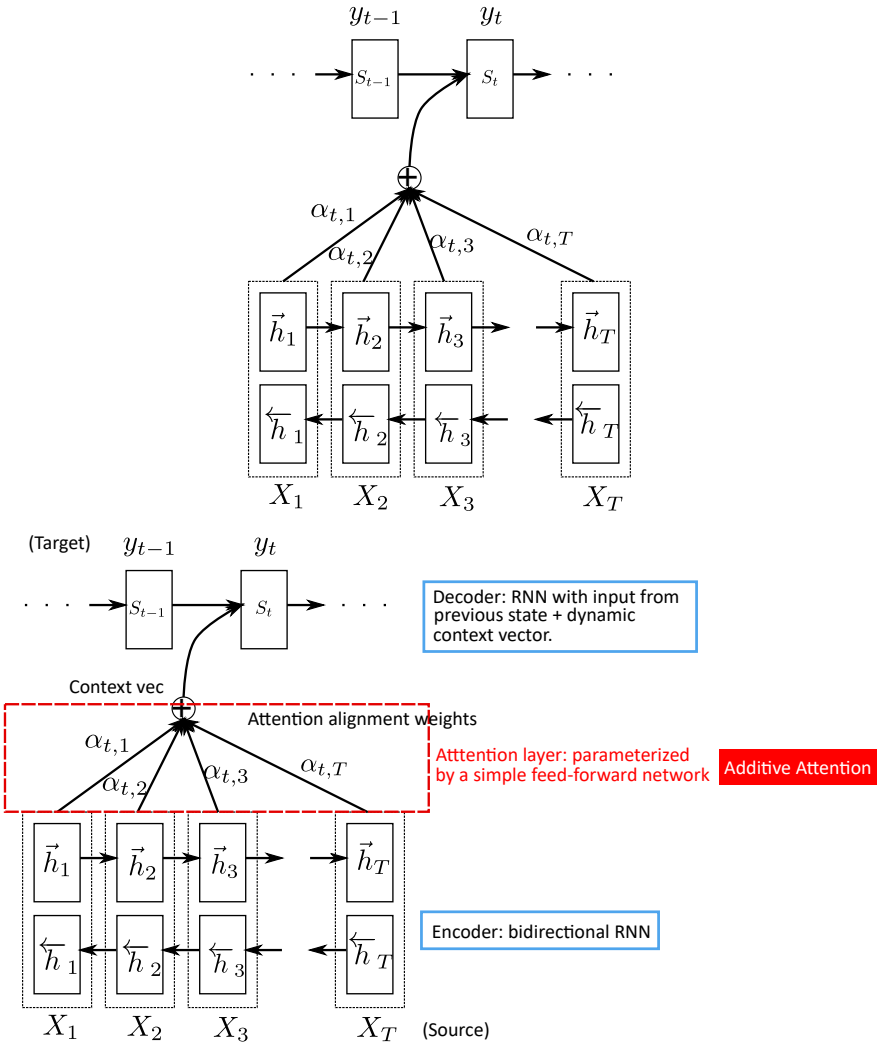

图 6：The encoder-decoder model with additive attention mechanism in Bahdanau et al., 2015.

Attention Model Definition

我们有一个长度为 n 的源序列 $\boldsymbol{x}$，并尝试输出长度为 m 的目标序列 $\boldsymbol{y}$：

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \tag{1}$$

$$\mathbf{y} = [y_1, y_2, \ldots, y_m] \tag{2}$$

（粗体变量表示它们是向量; 本文中其他所有内容都相同。）

编码器是双向 RNN（或您选择的其他循环网络设置），具有前向隐藏状态 $\overrightarrow{\boldsymbol{h}}_i$ 和后向隐藏状态 $\overleftarrow{\boldsymbol{h}}_i$。两个方向隐藏状态的简单“连结”表示一个编码器状态。这样做的动机是因为在一个单词的标注中包括前面和后面的单词。

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i^\top; \overleftarrow{\boldsymbol{h}}_i^\top]^\top, i = 1, \ldots, n$$

解码器网络具有隐藏状态 $\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}, y_{t-1}, \mathbf{c}_t)$ 用于在时刻 t 处来输出单词，$t = 1, \ldots, m$ 其中上下文向量 $\boldsymbol{c}_t$ 是输入序列的隐藏状态的总和，由联配值 (alignment scores) 加权得到：

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i \quad \text{; Context vector for output } y_t \tag{3}$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) \quad \text{; How well two words } y_t \text{ and } x_i \text{ are aligned.} \tag{4}$$

$$= \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))} \quad \text{; Softmax of some predefined alignment score..} \tag{5}$$

配分模型基于它们匹配的程度，将得分 $\alpha_{t,i}$ 分配给位置 i 处的输入对和位置 t 处的输出 (y_t, x_i)。$\{\alpha_{t,i}\}$ 的集合是定义每个输出应考虑每个源隐藏状态的多少的权重。在 Bahdanau 的论文中，联配值 $\boldsymbol{\alpha}$ 由**具有单个隐藏层的前馈神经网络**来进行参数化学习，并且该网络与模型的其他部分共同训练。因此，得分函数采用以下形式，因为 tanh 用作非线性激活函数：

$$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i]) \tag{6}$$

其中 $\mathbf{v}_a$ 和 $\mathbf{W}_a$ 都是在联配模型中学习的权重矩阵。

联配值矩阵是一个很好的副产品，可以明确显示源词和目标词之间的相关性。如下图所示：

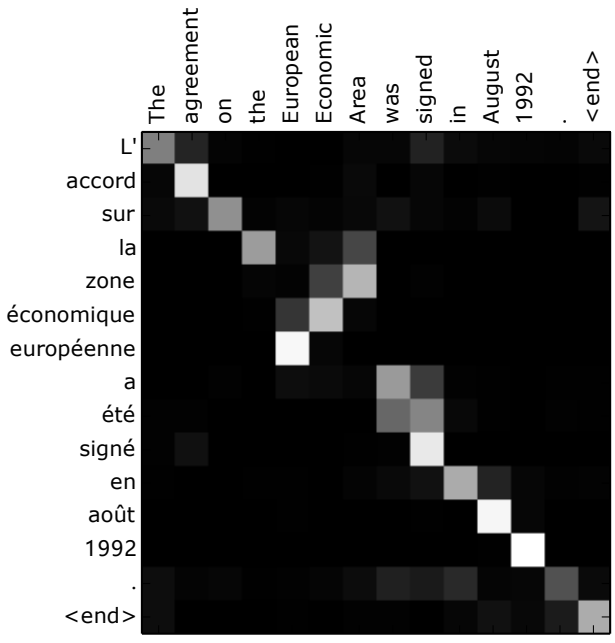

图 8：Alignment matrix of “L’accord sur l’Espace économique européen a été signé en août 1992” (French) and its English translation “The agreement on the European Economic Area was signed in August 1992”. (Image source: Fig 3 in Bahdanau et al., 2015.)

Attention 模型设计

在时间步 t，设解码器的背景变量为 $c_{t'}$，输出 $y_{t'}$ 的特征向量为 $\boldsymbol{y}_{t'}$。和输入的特征向量一样，这里每个输出的特征向量也是模型参数。解码器在时间步 t 的隐藏状态

$$\boldsymbol{s}_{t'} = g(\boldsymbol{y}_{t'-1}, \boldsymbol{c}_{t'}, \boldsymbol{s}_{t'-1}).$$

令编码器在时间步 t 的隐藏状态为 $\boldsymbol{h}_t$，且总时间步数为 T。解码器在时间步 t 的背景变量为

$$\boldsymbol{c}_{t'} = \sum_{t=1}^{T} \alpha_{t't} \boldsymbol{h}_t,$$

其中 $\alpha_{t't}$ 是权值。也就是说，给定解码器的当前时间步 t，我们需要对编码器中不同时间步 t 的隐藏状态求加权平均。这里的权值也称注意力权重。它的计算公式是

$$\alpha_{t't} = \frac{\exp(e_{t't})}{\sum_{k=1}^{T} \exp(e_{t'k})},$$

其中 $e_{t't} \in \mathbb{R}$ 的计算为

$$e_{t't} = a(\boldsymbol{s}_{t'-1}, \boldsymbol{h}_t).$$

上式中的函数 a 有多种设计方法。Bahanau 等使用了多层感知机：

$$e_{t't} = \boldsymbol{v}^\top \tanh(\boldsymbol{W}_s \boldsymbol{s}_{t'-1} + \boldsymbol{W}_h \boldsymbol{h}_t),$$

其中 v、$\boldsymbol{W_s}$、$\boldsymbol{W_h}$ 以及编码器与解码器中的各个权重和偏差都是模型参数。

Bahanau 等在编码器和解码器中分别使用了门控循环单元。在解码器中，我们需要对门控循环单元的设计稍作修改。解码器在 t' 时间步的隐藏状态为

$$\boldsymbol{s}_{t'} = \boldsymbol{z}_{t'} \odot \boldsymbol{s}_{t'-1} + (1 - \boldsymbol{z}_{t'}) \odot \tilde{\boldsymbol{s}}_{t'},$$

其中的重置门、更新门和候选隐含状态分别为

$$\begin{aligned}
\boldsymbol{r}_{t'} &= \sigma(\boldsymbol{W}_{yr} \boldsymbol{y}_{t'-1} + \boldsymbol{W}_{sr} \boldsymbol{s}_{t'-1} + \boldsymbol{W}_{cr} \boldsymbol{c}_{t'} + \boldsymbol{b}_r),\\
\boldsymbol{z}_{t'} &= \sigma(\boldsymbol{W}_{yz} \boldsymbol{y}_{t'-1} + \boldsymbol{W}_{sz} \boldsymbol{s}_{t'-1} + \boldsymbol{W}_{cz} \boldsymbol{c}_{t'} + \boldsymbol{b}_z),\\
\tilde{\boldsymbol{s}}_{t'} &= \tanh(\boldsymbol{W}_{ys} \boldsymbol{y}_{t'-1} + \boldsymbol{W}_{ss} (\boldsymbol{s}_{t'-1} \odot \boldsymbol{r}_{t'}) + \boldsymbol{W}_{cs} \boldsymbol{c}_{t'} + \boldsymbol{b}_s).
\end{aligned}$$

Transformer 架构

Transformer Attention

Bert Model

XLNet

Transformer XLNet

Reference