Attention注意力机制介绍

什么是Attention机制

Attention机制通俗的讲就是把注意力集中放在重要的点上，而忽略其他不重要的因素。其中重要程度的判断取决于应用场景，拿个现实生活中的例子，比如1000个人眼中有1000个哈姆雷特。根据应用场景的不同，Attention分为空间注意力和时间注意力，前者用于图像处理，后者用于自然语言处理。本文主要介绍Attention机制在Seq2seq中的应用。

为什么要用Attention机制

我们知道在Seq2seq模型中，原始编解码模型的encode过程会生成一个中间向量C，用于保存原序列的语义信息。但是这个向量长度是固定的，当输入原序列的长度比较长时，向量C无法保存全部的语义信息，上下文语义信息受到了限制，这也限制了模型的理解能力。所以使用Attention机制来打破这种原始编解码模型对固定向量的限制。

Attention原理

Attention的原理就是计算当前输入序列与输出向量的匹配程度，匹配度高也就是注意力集中点其相对的得分越高。其中Attention计算得到的匹配度权重，只限于当前序列对，不是像网络模型权重这样的整体权重。

算法过程:

1）encode对输入序列编码得到最后一个时间步的状态c，和每个时间步的输出h，其中c又作为decode的初始状态z0。

2）对于每个时间步的输出h与z0做匹配也就是match操作，得到每个时间步的匹配向量α01，如图1。

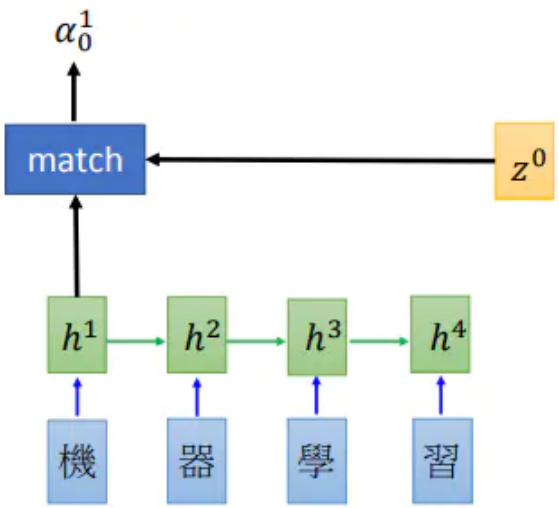

图1

3）对所有时间步的输出h与z0的匹配度α0，使用softmax做归一化处理，得到各个时间步对于z0的匹配分数。

4）求各个时间步的输出h与匹配分数的加权求和得到c0，作为decode的下一个时间步的输入，如图2。

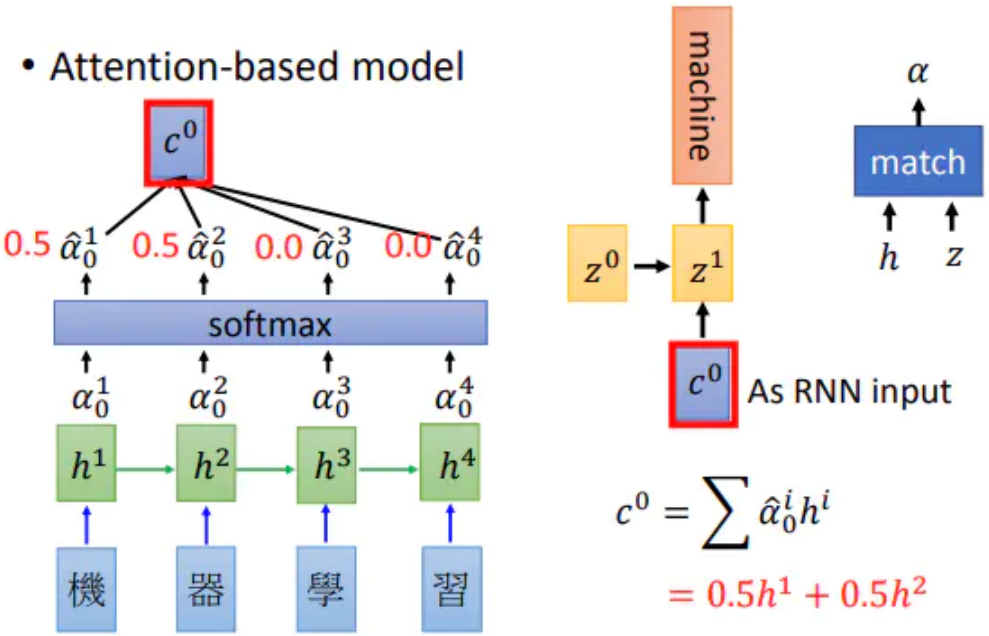

图2

5）计算各个时间步的输出h与z1的匹配度得到c1作为decode下一个时间步的输入，如此一步一步重复下去，如图3。

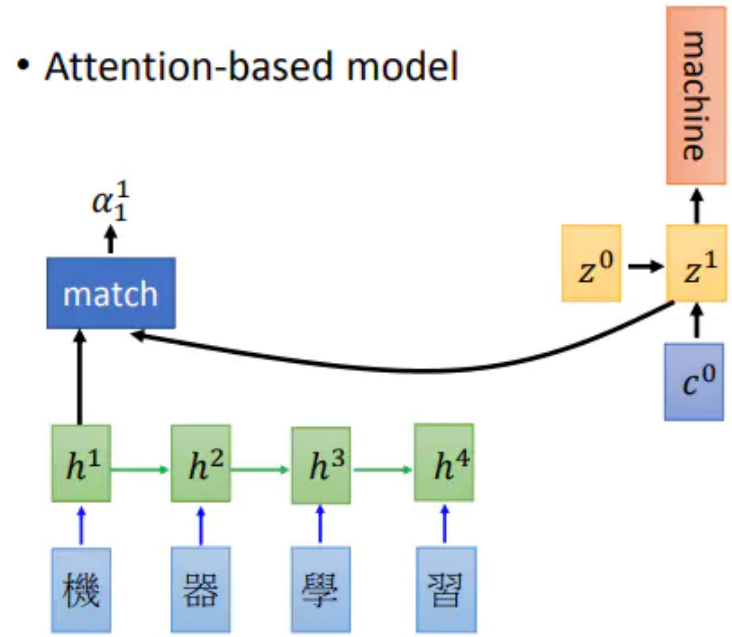

图3

这样就可以把每个时间步重要的信息传给decode中，以上就是Attention机制的处理过程。其中match操作一般是求两个向量的相似度，通常有如下方法：

1）余弦相似度

2）一个简单的 神经网络，输入为hh和ww，输出为α

3）或者矩阵变换α=hTWzα=hTWz (Multiplicative attention，Luong et al., 2015)

在tensorflow1.0版本以后的api seq2seq库中，包含了两种Attention算法，他们的区别就是match操作的不同，因此也有人称他们为加法Attention和乘法Attention，具体内容下：

1）BahdanauAttention：论文https://arxiv.org/abs/1409.0473中的实现：

$$u^t = v^T tanh(W_1 h + W_2 d_t)$$
$$a^t = softmax(u^t)$$
$$c^t = \sum_l^L a_l^t h_l$$

图4

2）LuongAttention：论文https://arxiv.org/abs/1508.04025中的实现：

$$u^t = d_t W_1 h$$
$$a^t = softmax(u^t)$$
$$c^t = \sum_l^L a_l^t h_l$$

图5

由于图片来自不同地方，所以符号有些不同，图4和图5中的h是上文所说的每个时间步的输出向量，d是decode中每个时间步的状态，也就是上文中的z，c是match后计算的权值加和后的向量用于decode中每个时间步的输入，a就是match操作中经过softmax后的匹配权重，v是一个向量，相当于w一样的权重需要去学习。有上面两个公式可以看出，BahdanauAttention和LuongAttention的区别就是在match过程中的计算方式不同，一个是将decode的状态与encode的输出求和，一个是求乘，所以才有了加法Attention和乘法Attention的叫法。

========== 更新 ==========

最近一段时间的学习,发现Attention的各种形式与用法,但是归根结底,都是同一种形式---Google的一般化Attention.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

图6

其中Q是query,K和V是一一对应的,相当于Key-Value的关系.一般情况下都会说求谁对谁的Attention,比如上面提到的Seq2seq中,decoder的隐状态z对encoder输出h的attention,那么隐状态z就相当于该式中的query,encoder的输出h就是key和value(这里key和value相等,也有不等的情况).所以如果说A对B的attention,那么A就是query,B就是key-value(key-value怎么分配看实际情况).这样,上文提到的内容就可以很容易的代入到一般会Attention中.先用query(decoder隐藏状态z)和key(encoder输出h)做点乘然后归一化,使用softmax计算权

重得分,再与value(encoder输出h)相乘得到最后的向量.(attention的机制像极了key-value记忆网络的原理, 或者更准确的说是key-value记忆网络像极了attention的机制,使用query与key做匹配运算,求得相关度得分,然后使用该得分与value运算,得到最后的向量).

特别注意的是,如果Q,K,V的值都是一个的话,那么就称为Self Attention.

参考:

台大李宏毅课程