**读书报告**

王贵涛

自然语言处理的第八章主要介绍了编解码器模型、注意力和上下文嵌入。

**编解码器网络（Encoder-Decoder Networks）**

上下文编码影响会随着输出序列的生成而减弱，一种解决方案是在每一步都提供上下文向量。输出层根据新的隐藏状态，前一状态输出和上下文编码来设置输出。

对于神经生成，从随机采样不合适，应该选择在每一个时间步骤选择最有可能的输出。不幸的是，动态规划不变量在这里不成立。

可以将解码问题是我状态空间搜索，使用定向搜索（Beam Search），通过将宽度优先搜索策略和启发式筛选器相结合，对每个选项评分，保留高分选择来减少空间。完成的假设可能有不同的长度，较短的假设评分倾向于会更高。通常的解决方法是对每个假设进行长度归一化。

上下文向量（context）被定义为编码器隐藏状态的函数，可隐藏状态的数量随着输入的大小而变化，使得直接使用它们解码非常困难。之前的方法将上下文向量减少到一个固定长度的向量，不过这个最终隐藏状态不可避免的更多关注输入序列的后一部分。或者使用Bi-RNNs，上下文向量是可以向前和向后传递最终状态的函数。或者简单地对编码器隐藏状态求和或平均，但丢失了许多有用的信息。

**注意力（Attention）**

为了克服简单上下文方法的不足，需要一种考虑整个编码器上下文的机制，在编码过程中动态更新，大小固定，这种方法称为注意力机制（attention mechanism）。

第一步先将静态上下文向量换成动态，在每一步解码中生成新的上下文向量。计算上下文向量需要计算一个分数向量，提供解码器隐藏状态与每个编码器隐藏状态的相似度，可以直接使用向量间点积。简单的点积不能适应，通过将得分用自己的一组权值参数化，可以得到更稳健的相似度评分。

将分数规格化以创建权重向量，最后根据权重向量的分布，对所以编码器隐藏状态的加权平均来计算当前解码器状态的定长上下文向量。

**编解码网络的应用（Applications of Encoder-Decoder Networks）**

对基本编码器网络的关注导致了性能的快速提高以广泛应用，包括总结，句子简化，问题回答和图像字母。