关于R-Net, FusionNet,QA-Net的理解与作业心得

1. R-Net

R-Net它的贡献在于将注意力融入RNN，同时又对向量进行了门控的思想。

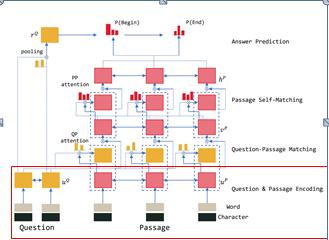
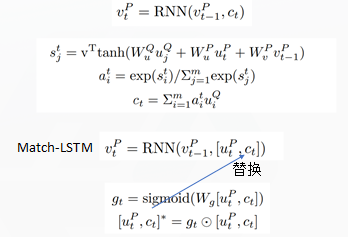


图1 结构

它包括四个层级

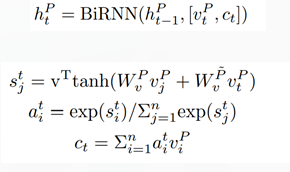
1.1、编码层它采用同样是Q,P是word char Embedding，经过双向的GRU,替代了LSTM，精简了参数。

1.2、交互层



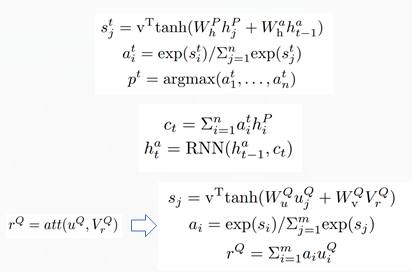
Ct的理解是将Q向量融入过来，求取P到Q的attention,但是此时这里是将t时候P的向量与Ct拼接，初始gt向量与其点积，强化了P向量，并做了门控的含义。Vtz则是上述的GRU的结果。这一步是将Q向量加进来同时加强了P向量。

1.3、self-Matching attention



它是对上一步的vt进行一个自我的attention。

1.4、输出层

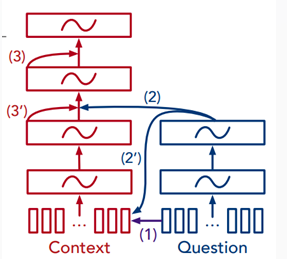


它又一次对Q做了attention，同时又与上一层计算得到ht计算相似矩阵

并做了softmax计算，然后求取开始与起始的位置。其中t只能取0,1,代表开始与结束位置。H0=R0,这一步同样加入了Q向量，加强了Q的理解。

1. FusionNet

FusionNet它的贡献在于提出了单词历史和全关注注意力。在我们进行单词高级层次向量时候，容易忘记低级的表示方式，从而无法获得单词的完整信息，单词历史这是将各个层级拼接，但是这样输出维度就会增加，降低效率，所以降低维度处理。全关注注意力则是对单词历史的向量进行Attention处理，里面对维度进行了压缩处理。



2.1、编码层

它来源于Glove，Cove，Pos,NER,Other向量总共922层

2.2 交互层

单词注意力层

它是单词到问题的注意力计算 922+300=1222层

阅读层

它是文章和问题的向量经过两层的LSTM,每一层都产生对应结果，分别输出250维向量。

阅读理解层

它是将上述两层问题向量做了拼接，然后经过双向的LSTM产生一个250向量

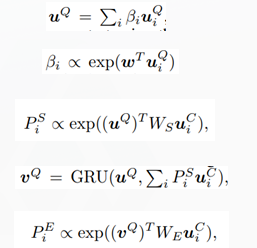
全关注互注意力层

它采用了单词历史的概念。Context的单词历史包括Glove，Cove,两层阅读层的向量组成，同理问题的单词历史也是类似，一共1400维向量，维度过高，要进行压缩，并且进行低层，高层，理解层的融合（问题对内容的融合）。也就是对单词和问题历史做相似矩阵并分别对两层问题阅读层的向量加权求和，和对问题阅读理解层的加权求和

全关注互注意力层

它是将context的两层向量以及上述三次的融合向量拼接起来再通过双向的LSTM

2.3输出层

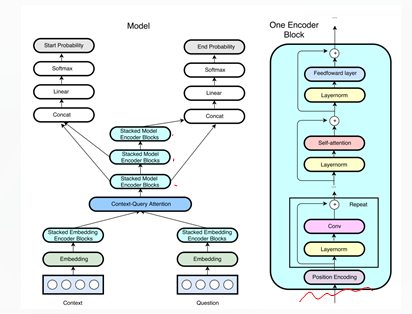


其中Uq是问题的理解向量，Uc是上面交互层最终产生的文章向量，当我们计算完开始位置概率，再将其与问题向量融合然后获得结束位置的概率。

3 QA-Net

由于上述采用都是RNN,训练较慢，它采用的是CNN+Self-Attention

同时采用了机器翻译来实现数据扩增



它的架构跟Bi-DAF类似，输入层采用了Char embedding,并且采用了两层的Highway Network。

Embedding层包括了LayNorm(它是特征维度的一个归一化），卷积层，自注意力层，以及前向传播层。

Context-Query Attention层同意是对C2Q和Q2C的注意力的处理。

Model Encoder层跟Embedding一致，但是里面的权值可以共享。

输出层计算的计算方式跟Bi-DAF相同。

以上就是我对上一节课的理解。

作业理解

关于上回疑惑点GLove矩阵使用，它第一步一样调用，比我多了一步将词构造词典，从而数值化。本人之前采用是直接

embedding，然后跑起来容易内存溢出，老师采用的是先数值化，然后embedding的。在本次作业中，其实思想都知道，但是计算时候需要注意张量的各个shape。