**一、主要思想**

1、soft&hard attention中的attention是把CNN最后一个卷积层的feature map加权求和后，进行全连接和embedding输入LSTM；

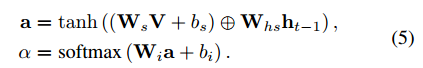
本文与此不同，本文的方法是直接修改CNN网络，具体方法是在CNN的不同layers，不同channels的feature map上乘一个权重，最后把全连接层的输出embedding后输入LSTM；

由此实现了spatial-channel-layer的attention；

2、由于同时学习spatial和channel的attention weights需要大量内存，对应一个t时刻和一个l卷积层，需要W\*H\*C的内存，故提出分开学习spatial和channel的方法；

3、**spatial attention**

对于一个layer的feature map，令V=[v1,v2,……,vm]，其中vi是c维向量，m=W\*H；



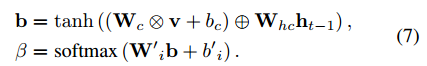
把c\*m维的V和d维的h映射到k\*m维的a，再用k\*m维的a计算m个spatial attention的权重alpha；

最终得到aplha是一个m维的weight向量；用于对m(W\*H)维的feature map加权；

4、**channel attention**

对于一个layer的feature map，令U=[u1,u2,……，uc]，其中ui是W\*H维向量；

令v=[v1,v2,…，vc],v是一个p\_=[(W/p)\*(H/p)]\*c维的向量，其中vi是ui的mean pooling，p是pooling filter的长度；



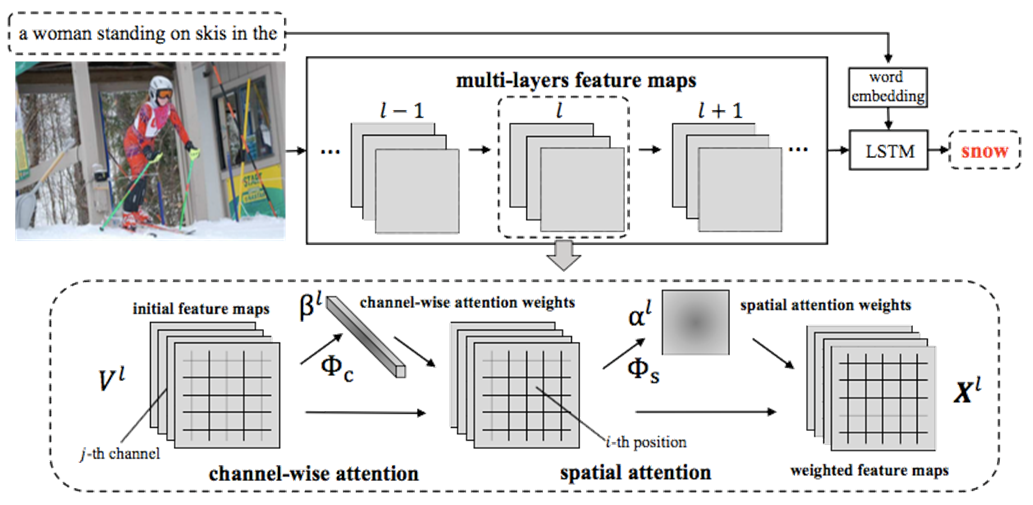
把p\_\*c维的v和d维的h映射到k\*c维的b，再用k\*c维的b计算c个channel attention的权重beta；

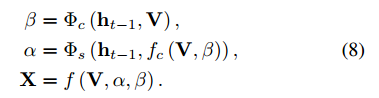
最终得到beta是一个c维的weight向量，用于对c维的feature map加权；

5、最终的应用方式是channel-spatial attention（如下所示）或spatial-channel attention：

channel-spatial attention是先对V求channel-wise的beta参数，用beta对V加权，记为V'，

再对V'求spatial-wise的alpha参数，用alpha对V'加权，记为V'',V''经过全连接层和embedding后输入LSTM;





C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(12).png

**二、疑问**

跑代码的时候注意一下alpha和beta的计算，因为本文的公式表达不清晰；

**三、idea**

本文是从attention的角度改进CNN；