本周主要工作：

上次是学到了提出了对应自编码器来表示两个单模态数据的公共部分，那么接下来还是接着学习这个模型。

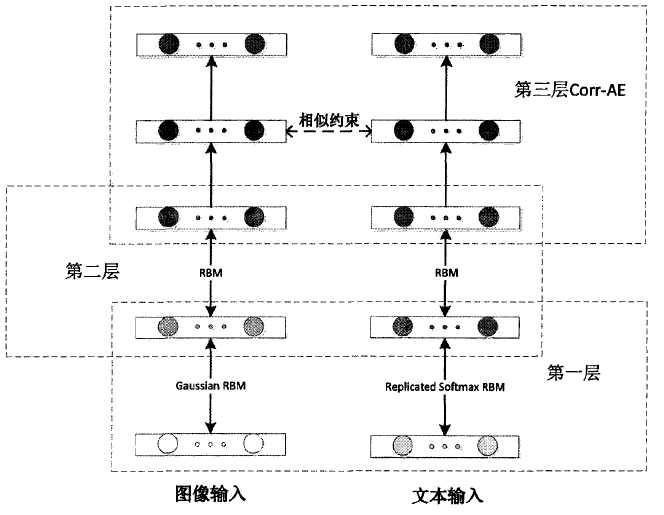
这里，作者提到浅层的模型不足以学习不同模态数据之间的关联，为此提出了深层结构。

深层结构：

特点：三层网络

第一二层是RBM（限制玻尔兹曼机），第三层是Corr-AE（对应自编码器）。第一层是扩展玻尔兹曼机，对于文本模态，其输入特征是离散值，故用RSRBM（针对离散值的），对于图像模态，其输入是连续的实数值，故用GRBM（针对实数值的）；那么在学习完了第一层的特征后，再把第一层的输出作为第二层的输入，构建更高层的两个单模态数据的表示。然后再把第二层的输出作为第三层的输入，主要作用是负责两个不同模态数据之间的关联性学习。

模型概念图：



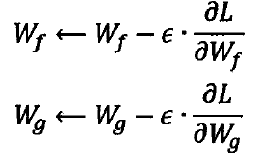
数据的训练过程是用深度结构训练算法实现的

文中对于这个算法的描述如下：

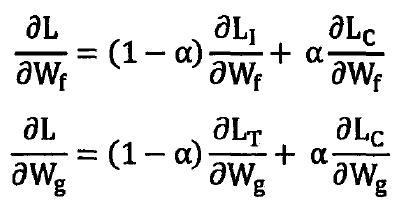
1. 利用对比散度算法算法训练第一层的GRBM和RSRBM；
2. 利用对比散度算法训练第二层的两个基本RBM；

（对比散度算法用前面讲过的CD-k抽样算法）

1. 利用梯度下降算法更新第三层网络，Corr-AE的参数，如下：



Ɛ是学习速率，L对权重的偏导数由下列公式来计算：



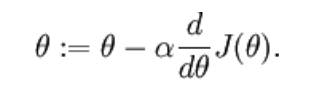
α是两个模态数据之间的平衡参数。

1. 一直重复1、2、3步骤，直至收敛

当深层网络训练完毕后，图像和文本这两个不同模态会被子网络映射到同一个表示空间中。

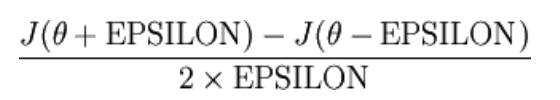
这时，若输入一个文本，那么这个数据就会和所有的图片数据映射到同一个表示空间中。这样，就可以用简单的距离函数来衡量两个模态的相似度。

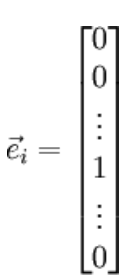
还学习了梯度检验与高级优化，主要是讲解当使用反向传播时存在bug时如何去发现程序的错误（一般是索引的缺位错误，例如漏了计算偏置项）。此方法是利用导数的的定义来验证求导代码的正确性。



这个公式是我们经常用来算梯度下降的公式，它是利用了求导的数学定义来算出一个极限来逼近我们代码算的导数来验证求导数的代码是否正确。

用以下这个公式来模拟求导数：



那么，当EPSILON为10-4时，这个结果至少和代码求出的导数有4位有效数字是一样的。但是如果IMG_256是一个向量时，就要变化一下，，，，那么仍然可以用这个公式来验证求导是否正确，