受限玻尔兹曼机（RBM）模型解释及代码实现

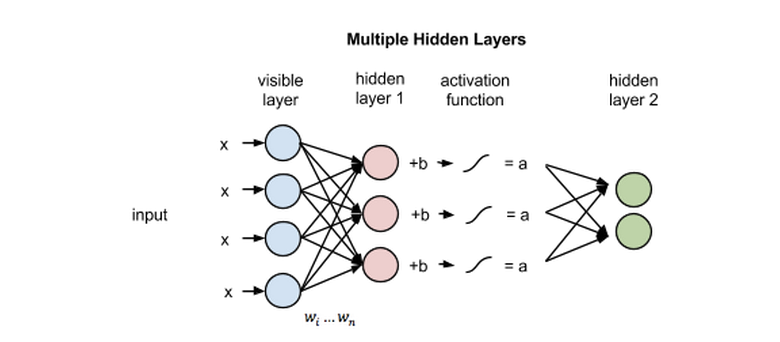
主要针对深度学习工具包的几个例子，NN（神经网络）和SAE（自编码模型，见文件夹下PPT和代码说明以及MATLAB16版之后**trainAutoencoder**函数）在这里就不再赘述。这里主要说明RBM（受限玻尔兹曼机）的原理和代码说明。CNN（卷积神经网络）这里也不做说明，因为模型比较复杂，参数过多，要求样本量比较大（不太适合实验室现有的磁共振数据），MATLAB的运行效率低，且该工具包的CNN模型拓展性比较差，CNN模型可以在python环境下的tensorflow（及其他平台下）实现简单高效的实现（或者16版及之后的matlab查询**trainNetwork**即可**参数说明在后面**）。

**注意**：神经网络参数多且复杂，模型的调参很重要。调参之前最好对要做分类（回归）的特征大致进行检查，看是否具备分类（回归）的能力，避免陷入死循环（可能特征就不具备分类的或者回归的能力）。

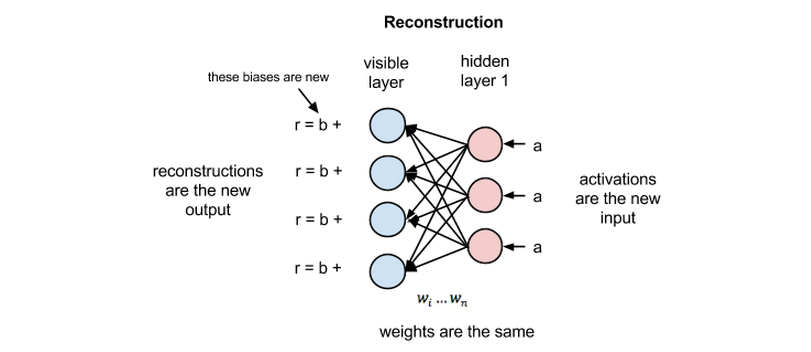
**受限玻尔兹曼机**

受限玻尔兹曼机（RBM）由Geoff Hinton发明，是一种用于降维、分类、回归的算法。和传统的神经网络方法相比最主要的区别它是基于概率模型计算和更新权重的。

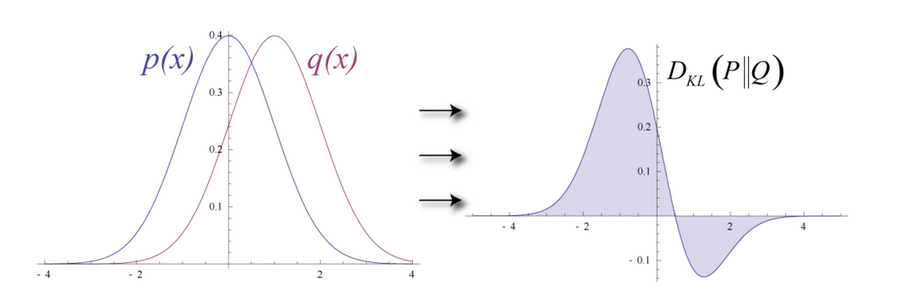
学习的过程可以分为两步（类比于BP网络）：前向算法和后向算法。前向算法：



后向算法：这里的后向算法是跟前向算法类比的，不同于BP算法后向传播的是误差，这里的后向传播指的是从输出到输入的过程。



因为刚开始的权重是随机的，前向传播和后向传播的结果肯定是不一致的。用概率表示就是p(a|x; w)，意思是基于输入x和权重w得到的a值的分布概率。p(x|a; w) 意思是基于输出a和权重w得到的输入值x的分布概率。刚开始（权重随机的时候）这两个概率分布肯定是差异很大的。要求的最终目的是使得两者的联合概率值最大p(x, a)。这样的算法也称之为重构算法。RBM使用KL散度来表示预测的概率密度分布和输入值的分布之间的距离的。如下图所示：



其中，p(x)和q(x)分别表示输入和输出的概率密度分布函数，而右边的图的阴影部分表示两者的KL散度。权重更新的目的就是为了让阴影下的面积值最大。

理解（直观上地）：类似于我们做模型预测的时候，经常把预测值和实际值做相关，相关系数高的认为是预测好的。这里的思想类似，训练的目的并不是为了精准的得到每一个输出值，而是可以很好地预测输出值得概率密度分布。传统的学习总是渴求得到尽量精确的预测值。

参考教程：1，<https://deeplearning4j.org/cn/restrictedboltzmannmachine> 基础讲解的很清楚（科普性质的）。

2，<http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/> 讲解的比较专业，包括权重的更新策略以及python的代码实现。

**代码说明**：RBM1里面用的是早发精分全脑功能连接（AAL90）数据。参数以及含义见代码注释。

**CNN MATLAB函数参数大致说明（推荐使用tensorflow，因为这个模型复杂了会特别慢）**：根据设计好的网络架构（比较死板，一般是固定的）在函数里面添加合适的层即可。下面的这个例子是MALTAB 2016自带例子的一个结构。

layers = [imageInputLayer([28 28 1],'Normalization','none');%输入28\*28图片。对输入进行标准化。

convolution2dLayer(5,20);%卷积层

reluLayer();%激活函数，类似于sigm函数（）

maxPooling2dLayer(2,'Stride',2);%池化层（降维）

convolution2dLayer(5,16);

reluLayer();

maxPooling2dLayer(2,'Stride',2);

fullyConnectedLayer(256);

reluLayer();

fullyConnectedLayer(10);

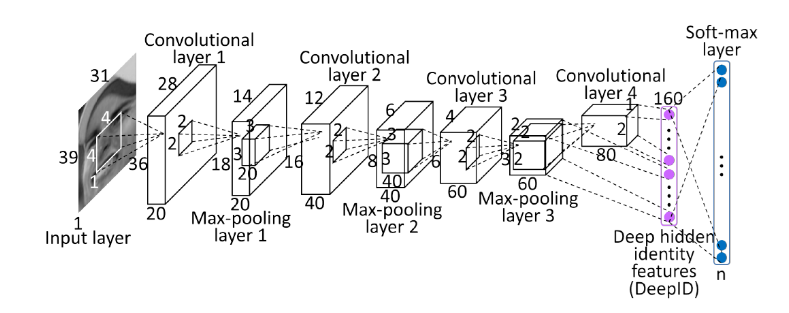
softmaxLayer();%分类函数

classificationLayer()];

设计好之后用trainingOptions和trainNetwork训练即可（具体参见函数说明）。

从例子可以看出，CNN的结构（目前的结构都是根据前人文章设计好的结构模型或者加少许修改得到的）一般是若干个{卷积层（包括激励函数）+ 池化层}，以及最后的全连接层（可以理解为传统的神经网络层）还有分类层（分类函数例如多分类的softmax（logisit回归的多分类版））构成。

参数很庞大，模型比较固定，需要大样本来支持。



完整的演示例子在CNN1里面（用MATLAB16之后的版本跑）CNN1，跑起来需要三分钟左右。