DL012(2016)

DL013(2018)

DL018(2016)

DL017(2016)

DL009(2016)

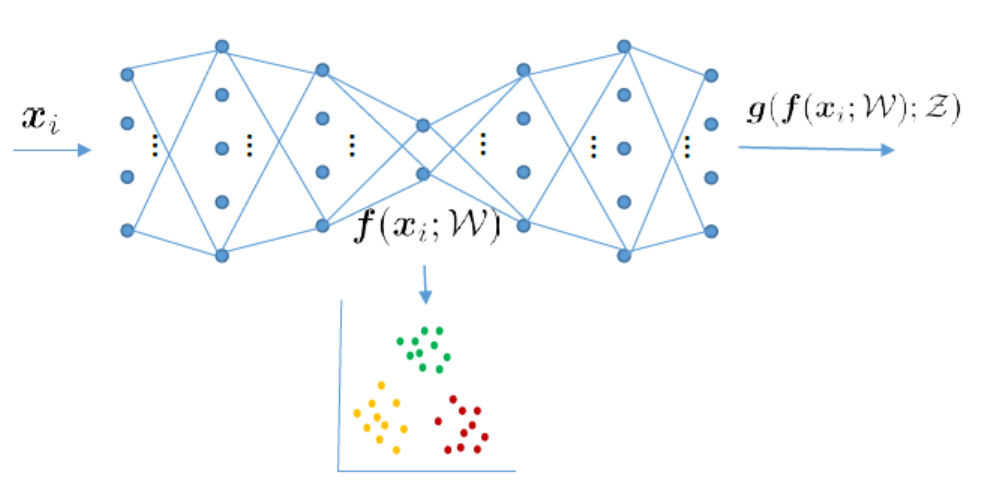
DL016(2016)

**箭头表示被引用，例如DL016被DL012引用**

# DL012 Towards Kmeans-friendly Spaces Simultaneous Deep Learning and Clustering (2016 ICML)

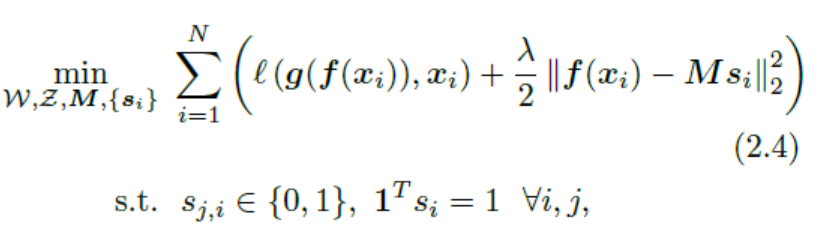
## 1 网络结构：

也是自编码器，聚类的事情在编码完成就做了，后面的解码和聚类无关，但是和降维有关。



## 2 工作原理和目标函数：

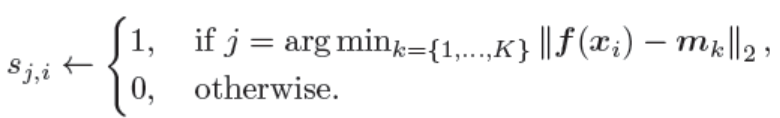
把DL016的线性变换变成了非线性变换f，损失函数l是为了让变换以后的f(x)可以尽量恢复为x，这就是自编码器的工作目标啊，聚类项是均方误差，这个和k均值聚类一样。M是聚类中心的矩阵，s是样本的聚类指示向量。不过我就是不明白，用神经网络干这件事，怎么能够限定s为0-1变量呢？如何做到？



## 3 优化方法：

交替迭代

Step1 固定M和s，优化W与Z就是自编码器加上一个和聚类性能有关的惩罚项，用SGD就可以；

Step2 固定M和W、Z，优化s采用的是

哦，原来是这样保障s是0-1的，我就说嘛，用神经网络怎么能做这件事啊

Step3 固定W、Z和s，优化M就是求聚类中心，这个方法就多得是了。

所以说来说去，聚类这个工作还是不是由神经网络来做的，这也是正常的。只要聚类和降维同时优化就已经是很大的进步了。

## 4 数据集和评价指标：

指标NMI、ARI、ACC；数据集RCV1、20Newsgroup、Raw MNIST、Pre-Processed MNIST

## 5 对比的方法：

Kmeans、Spectral Clustering、Sparse Subspace Clustering with Orthogonal Matching Pursuit(SSC-OMP)、Local Consistent Concept Factorization、XRAY、NMF followed by K-means、Stacked Auto encoder followed by K-means、Joint NMF and K-means，哇塞，和这么多方法进行了比较啊

## 6 源码和环境：

<https://github.com/boyangumn/DCN>，开发环境不知道，我没打开看

## 7 创新点：

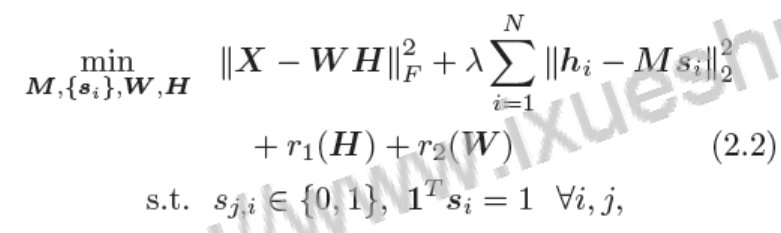
把深度网络做降维和聚类同时训练，用聚类的性能和降维后的可重构性作为优化目标指导深度网络训练，与DL009的区别是DL009先把自编码器训练得差不多了，然后再纯粹根据聚类性能来修正网络参数，而DL012是把聚类性能和自编码的性能作为优化目标同时训练网络参数。

# DL016 Learning from hidden traits Joint factor analysis and latent clustering (2016 TSP)：

闹了半天，原来不是用深度网络实现的啊，我就说嘛，线性的降维完全可以通过交替迭代进行优化啊，何必用神经网络?

## 2 工作原理和目标函数：

降维和聚类同时进行，下面的目标函数中，第一项是降维，第二项是聚类，就是最小均方误差，和K均值聚类是一样的，后面两项是正则化项。X是原始数据矩阵，H是降维以后的数据矩阵，M是聚类中心的矩阵，s是样本的聚类指示向量。看起来和小陶做的那个好像啊。



# DL009 Unsupervised Deep Embedding for Clustering Analysis (2016 ICML)

## 1 网络结构：

自编码器的编码部分

## 2 工作原理和目标函数：

Step1 给定初始聚类中心uj和网络参数theta，把样本xi映射到zi，根据zi到各uj的距离分配每个样本给各聚类中心，是软分配，不是0-1硬分配。软分配的结果是qij，即第i个样本属于第j个聚类的概率。这个分配过程有公式，可以用网络直接计算出来，这一部分的网络参数不需要学习。

Step2 学习参数theta和聚类中心u，目标函数是p和q的KL散度L。理想的分布p是冲击函数，实际中p也要比较尖锐，就取离聚类中心最近的几个样本z计算出p。

以上两步交替进行。

## 3 优化方法：

带动量的随机梯度下降（SGD），梯度用BP算法求。目标函数里有u有z，L先对z求导，然后反向传播得到L对theta的导数。

## 4 数据集和评价指标：

MNIST、STL-10、REUTERS，指标就是聚类的准确率，clustering accuracy，这显然是有监督的指标了。

## 5 对比的方法：

kmeans、LDMGI（2010 TIP）、SEC（2011 TNN）后两个都是基于Laplacian矩阵的，比的论文档次很高

## 6 源码和环境：

cafe的开发环境，不知道好不好移植到pytorch上，源码在<https://github.com/piiswrong/dec>

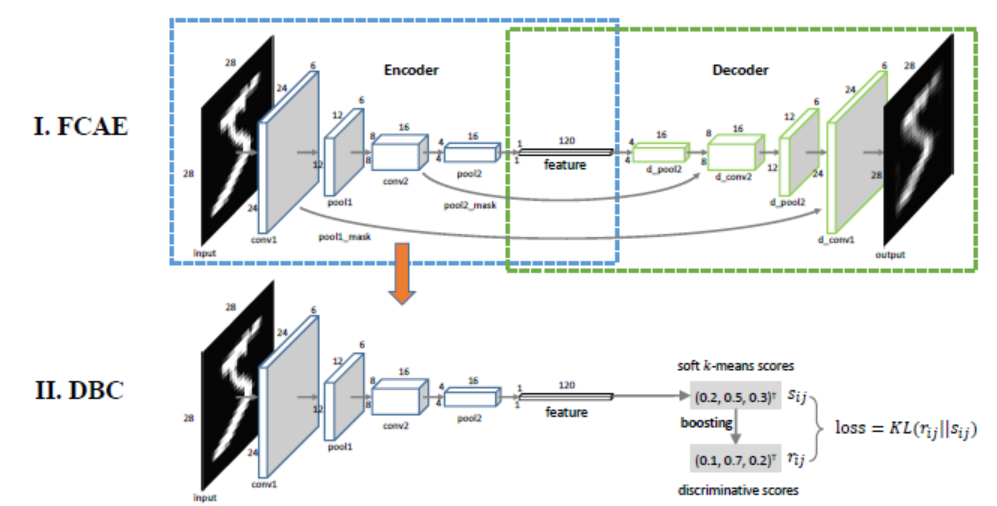
## 7 创新点：

把深度网络做降维和聚类同时训练，用聚类的性能为优化目标指导深度网络训练

# DL013 Discriminatively Boosted Image Clustering with Fully Convolutional Auto-Encoders(2018 PR)

## 1 网络结构：

全卷积自编码器，训练完以后把编码部分拿过来



## 2 工作原理和目标函数：

工作原理和DL009好像啊，训练好的自编码器作为初始值，然后交替迭代训练神经网络的参数和聚类中心，优化的目标函数也是KL散度，求导也求得完全一样，这样行不行啊？

## 3 优化方法：

就是梯度下降法

## 4 数据集和评价指标：

MNIST、USPS、 COIL-20、COIL-100；评价指标 ACC NMI

## 5对比的方法：

KMS、DAE-KMS、AEC、IEC DEC（就是DL009）、DCN、FACE-KMS、DBC

## 6 源码和环境：

貌似没有给源码

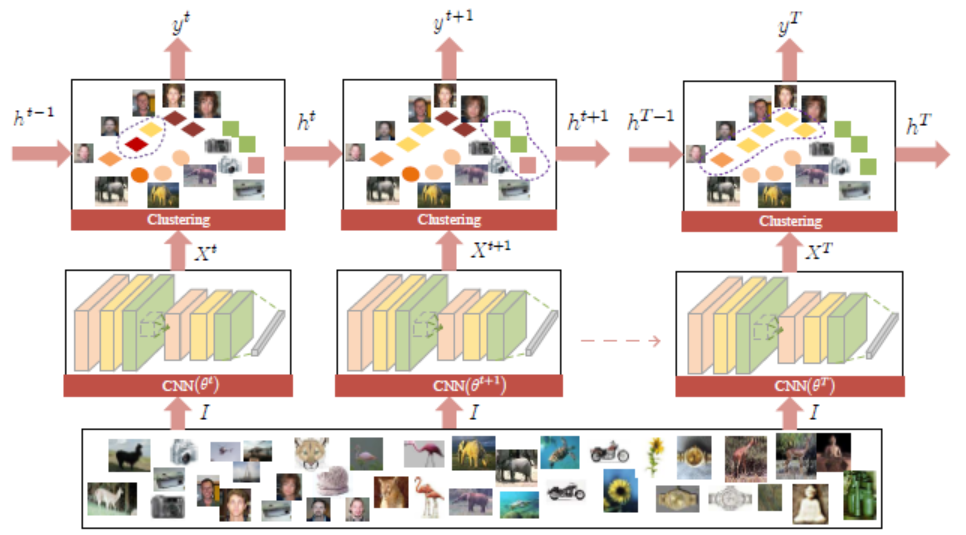
## 7 创新点：

感觉没啥创新，就是把DL009的稍微改了一下而已。

# DL017 Joint Unsupervised Learning of Deep Representations and Image Clusters (2016 CVPR)

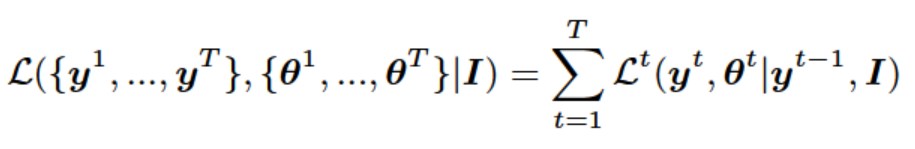
## 1 网络结构：

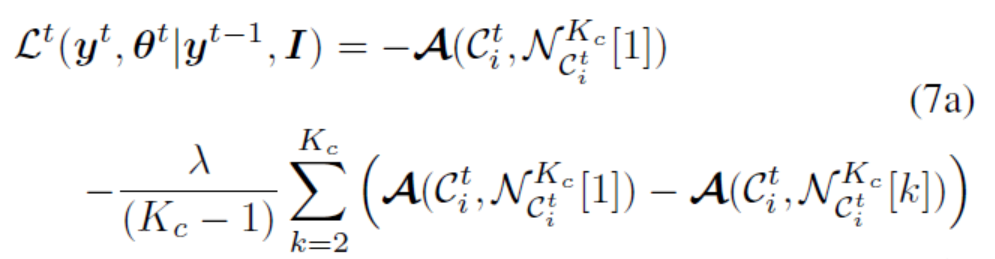
一系列CNN。当然了，是不是系列也不重要，反正这种迭代的优化方式本来也要训练多次神经网络，每一次就算一个神经网络。



## 2 工作原理和目标函数：

聚类采用的是层次聚类中的凝聚聚类，每次合并最像的两簇。也是迭代地交替优化，目标函数是





其中y是用层次聚类的结果，I是图像，\theta是CNN的参数，上标t是轮次。每一个轮次里面，固定y优化theta，然后固定theta优化y。7a公式的右端虽然没有显示地出现y和theta，但是y和theta决定了聚类结果，直接影响右端的值，不过写不出显示的表达式，该怎么求导呢？这里没搞清楚细节。

## 3 优化方法：

训练神经网络求theta就是梯度下降法，优化y就是合并最像的两个簇。

## 4 数据集和评价指标：

数据集是MNIST, USPS, COIL20, COIL100, UMist, FRGCV2.0, CMU-PIE, YTF。指标是NMI

## 5对比的方法：

K-means、SC-NJW、SC-ST、N-Cuts……共12种方法

## 6 源码和环境：

<https://github.com/jwyang/joint-unsupervised-learning>，环境是caffe

## 7 创新点：

感觉挺创新的，用了CNN和层次聚类，层次聚类本来就是一次一次聚的，比较适合用多次迭代的优化方式，目标函数用的是最大似然，和看过的几篇也不一样。

# DL018 Infinite Ensemble for Image Clustering (2016 KDD)

## 1 网络结构：

自编码器

## 2 工作原理和目标函数：

没有仔细看，只懂个大概。大致意思是自编码器的输入就是一个个的basic partitions，输出是一个统一的partition，目标函数就是让输入和输出一致，当然了，他定义了合适的度量来衡量两者的一致性，作为目标函数。

## 3 优化方法：

就是梯度下降法吧

## 4 数据集和评价指标：

13个实际图像的数据库:Letter, MINST, COIL100, Amazon, Caltech, Dslr, Webcam, ORL, USPS, Caltech101, ImageNet, Sun09, VOC2007。指标是accuracy 和NMI。

## 5对比的方法：

K-means， MAEC，GEncoder, DLC, GCC,HCC, KCC, SEC。好像没有和深度聚类的比，估计是同一时期出来的文章，来不及比。

## 6 源码和环境：

好像木有给。

## 7 创新点：

我觉得很创新啊，别人把自编码器当作是特征提取的手段，提取完特征再在深度网络之外做聚类操作，然后把聚类的性能作为损失函数来指导自编码器的学习过程。本文的不同在于他的聚类和特征提取真的都是在deep网络中完成的，这样的话也不需要迭代优化了吧？好像不对，他最后怎么也用了K-means啊？这不是成了最原始的那种先用深度网络做特征提取，然后再聚类的方法了吗？只是他提取特征的目标函数和别人的不一样而已啊。哎呀不懂了。