17 Sep

变分自编码器（四）：一步到位的聚类方案

By 苏剑林 | 2018-09-17 | 341958位读者 引用

由于VAE中既有编码器又有解码器（生成器），同时隐变量分布又被近似编码为标准正态分布，因此VAE既是一个生成模型，又是一个特征提取器。在图像领域中，由于VAE生成的图片偏模糊，因此大家通常更关心VAE作为图像特征提取器的作用。提取特征都是为了下一步的任务准备的，而下一步的任务可能有很多，比如分类、聚类等。本文来关心“聚类”这个任务。

一般来说，用AE或者VAE做聚类都是分步来进行的，即先训练一个普通的VAE，然后得到原始数据的隐变量，接着对隐变量做一个K-Means或GMM之类的。但是这样的思路的整体感显然不够，而且聚类方法的选择也让我们纠结。本文介绍基于VAE的一个“一步到位”的聚类思路，它同时允许我们完成无监督地完成聚类和条件生成。

理论

一般框架

回顾VAE的loss（如果没印象请参考《变分自编码器（二）：从贝叶斯观点出发》）：

$$KL\Big(p(x,z)\Big\Vert q(x,z)\Big) = \iint p(z|x)\tilde{p}(x)\ln \frac{p(z|x)\tilde{p}(x)}{q(x|z)q(z)}dzdx \tag{1}$$

通常来说，我们会假设$q(z)$是标准正态分布，$p(z|x), q(x|z)$是条件正态分布，然后代入计算，就得到了普通的VAE的loss。

然而，也没有谁规定隐变量一定是连续变量吧？这里我们就将隐变量定为(z,y)，其中z是一个连续变量，代表编码向量；y是离散的变量，代表类别。直接把(1)中的z替换为(z,y)，就得到

$$KL\Big(p(x,z,y)\Big\|q(x,z,y)\Big)=\sum_y\iint p(z,y|x)\tilde{p}(x)\ln\frac{p(z,y|x)\tilde{p}(x)}{q(x|z,y)q(z,y)}dzdx\tag{2}$$

这就是用来做聚类的VAE的loss了。

分步假设

啥？就完事了？呃，是的，如果只考虑一般化的框架，(2)确实就完事了。

不过落实到实践中，(2)可以有很多不同的实践方案，这里介绍比较简单的一种。首先我们要明确，在(2)中，我们只知道$\tilde{p}(x)$（通过一批数据给出的经验分布），其他都是没有明确下来的。于是为了求解(2)，我们需要设定一些形式。一种选取方案为

$$p(z,y|x)=p(y|z)p(z|x),\quad q(x|z,y)=q(x|z),\quad q(z,y)=q(z|y)q(y)\tag{3}$$

代入(2)得到

$$KL\Big(p(x,z,y)\Big\|q(x,z,y)\Big)=\sum_y\iint p(y|z)p(z|x)\tilde{p}(x)\ln\frac{p(y|z)p(z|x)\tilde{p}(x)}{q(x|z)q(z|y)q(y)}dzdx$$

其实(4)式还是相当直观的，它分布描述了编码和生成过程：

> 1、从原始数据中采样到x，然后通过$p(z|x)$可以得到编码特征z，然后通过分类器$p(y|z)$对编码特征进行分类，从而得到类别；
>
> 2、从分布$q(y)$中选取一个类别y，然后从分布$q(z|y)$中选取一个随机隐变量z，然后通过生成器$q(x|z)$解码为原始样本。

具体模型

(4)式其实已经很具体了，我们只需要沿用以往VAE的做法：$p(z|x)$一般假设为均值为$\mu(x)$方差为$\sigma^2(x)$的正态分布，$q(x|z)$一般假设为均值为$G(z)$方差为常数的正态分布（等价于用MSE作为loss），$q(z|y)$可以假设为均值为μ_y方差为1的正态分布，至于剩下

的$q(y), p(y|z)$，$q(y)$可以假设为均匀分布（它就是个常数），也就是希望每个类大致均衡，而$p(y|z)$是对隐变量的分类器，随便用个softmax的网络就可以拟合了。

最后，可以形象地将(4)改写为

$$\mathbb{E}_{x\sim\tilde{p}(x)}\left[-\log q(x|z)+\sum_y p(y|z)\log\frac{p(z|x)}{q(z|y)}+KL\big(p(y|z)\big\Vert q(y)\big)\right],\quad z\sim p(z|x$$

其中$z\sim p(z|x)$是重参数操作，而方括号中的三项loss，各有各的含义：

1、$-\log q(x|z)$希望重构误差越小越好，也就是z尽量保留完整的信息；

2、$\sum_y p(y|z)\log\frac{p(z|x)}{q(z|y)}$希望$z$能尽量对齐某个类别的“专属”的正态分布，就是这一步起到聚类的作用；

3、$KL\big(p(y|z)\big\Vert q(y)\big)$希望每个类的分布尽量均衡，不会发生两个几乎重合的情况（坍缩为一个类）。当然，有时候可能不需要这个先验要求，那就可以去掉这一项。

实验

实验代码自然是Keras完成的了（^_^），在mnist和fashion-mnist上做了实验，表现都还可以。实验环境：Keras 2.2 + tensorflow 1.8 + Python 2.7。

代码实现

代码位于：https://github.com/bojone/vae/blob/master/vae_keras_cluster.py

其实注释应该比较清楚了，而且相比普通的VAE改动不大。可能稍微有难度的是$\sum_y p(y|z)\log\frac{p(z|x)}{q(z|y)}$这个怎么实现。首先我们代入

$$\begin{aligned}p(z|x)=&\frac{1}{\prod\limits_{i=1}^{d}\sqrt{2\pi\sigma_i^2(x)}}\exp\left\{-\frac{1}{2}\left\Vert\frac{z-\mu(x)}{\sigma(x)}\right\Vert^2\right\}\\ q(z|y)=&\frac{1}{(2\pi)^{d/2}}\exp\left\{-\frac{1}{2}\Vert z-\mu_y\Vert^2\right\}\end{aligned}\tag{6}$$

得到

$$\log\frac{p(z|x)}{q(z|y)}=-\frac{1}{2}\sum_{i=1}^{d}\log\sigma_i^2(x)-\frac{1}{2}\left\Vert\frac{z-\mu(x)}{\sigma(x)}\right\Vert^2+\frac{1}{2}\Vert z-\mu_y\Vert^2\tag{7}$$

注意其实第二项是多余的，因为重参数操作告诉我们 $z=\varepsilon\otimes\sigma(x)+\mu(x),\ \varepsilon\sim\mathcal{N}(0,1)$，所以第二项实际上只是$-\Vert\varepsilon\Vert^2/2$，跟参数无关，所以

$$\log\frac{p(z|x)}{q(z|y)}\sim-\frac{1}{2}\sum_{i=1}^{d}\log\sigma_i^2(x)+\frac{1}{2}\Vert z-\mu_y\Vert^2\tag{8}$$

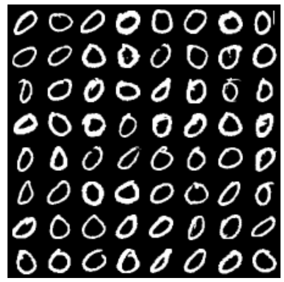
然后因为y是离散的，所以事实上$\sum_y p(y|z)\log\frac{p(z|x)}{q(z|y)}$就是一个矩阵乘法（相乘然后对某个公共变量求和，就是矩阵乘法的一般形式），用K.batch_dot实现。

其他的话，读者应该清楚普通的VAE的实现过程，然后才看本文的内容和代码，不然估计是一脸懵的。

mnist

这里是mnist的实验结果图示，包括类内样本图示和按类采样图示。最后还简单估算了一下，以每一类对应的数目最多的那个真实标签为类标签的话，最终的test准确率大约有83%，对比这篇文章《Unsupervised Deep Embedding for Clustering Analysis》的结果（最高也是84%左右），感觉应该很不错了。

聚类图示

			
聚类类别_0	聚类类别_1	聚类类别_2	聚类类别_3
			
聚类类别_4	聚类类别_5	聚类类别_6	聚类类别_7
			
	聚类类别_8	聚类类别_9	

按类采样


类别采样_0


类别采样_1


类别采样_2


类别采样_3


类别采样_4


类别采样_5


类别采样_6


类别采样_7

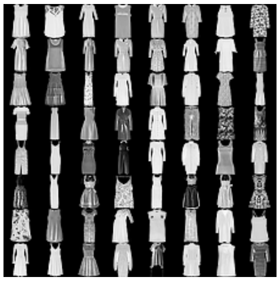
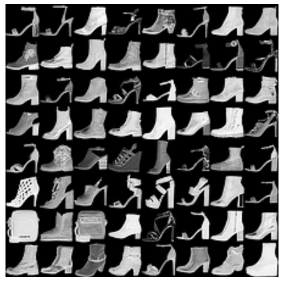
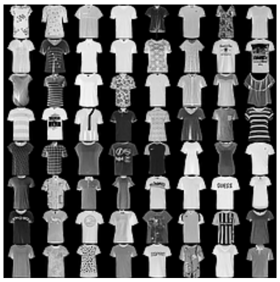
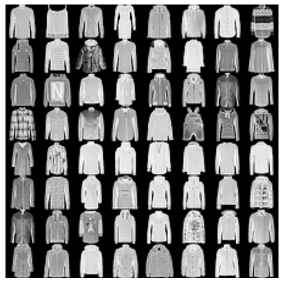
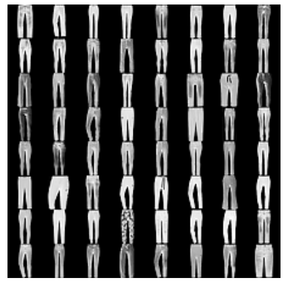
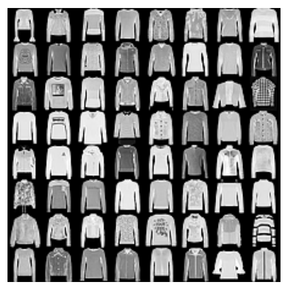

类别采样_8


类别采样_9

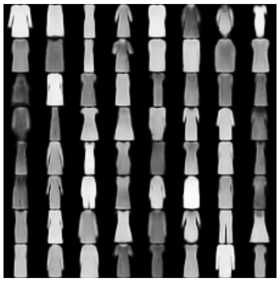
fashion-mnist

这里是fashion-mnist的实验结果图示，包括类内样本图示和按类采样图示，最终的test准确率大约有58.5%。

聚类图示

			
聚类类别_0	聚类类别_1	聚类类别_2	聚类类别_3
			
聚类类别_4	聚类类别_5	聚类类别_6	聚类类别_7
			
	聚类类别_8	聚类类别_9	

按类采样

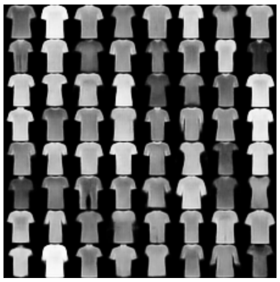


类别采样_0

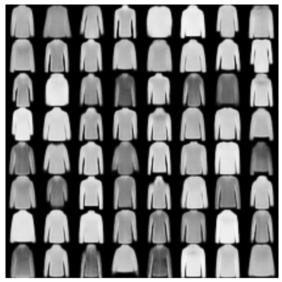
类别采样_1

类别采样_2

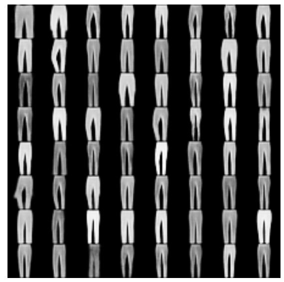
类别采样_3



类别采样_4

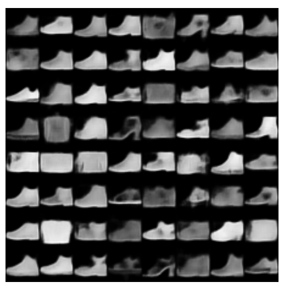


类别采样_5

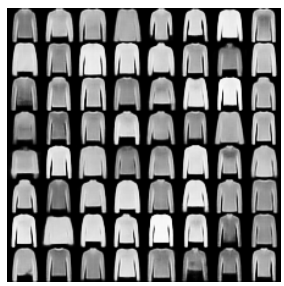


类别采样_6

类别采样_7



类别采样_8



类别采样_9

总结

文章简单地实现了一下基于VAE的聚类算法，算法的特点就是一步到位，结合“编码”、“聚类”和“生成”三个任务同时完成，思想是对VAE的loss的一般化。

感觉还有一定的提升空间，比如式(4)只是式(2)的一个例子，还可以考虑更加一般的情况。代码中的encoder和decoder也都没有经过仔细调优，仅仅是验证想法所用。

如果您需要引用本文，请参考：

苏剑林. (Sep. 17, 2018). 《变分自编码器（四）：一步到位的聚类方案 》[Blog post]. Retrieved from https://spaces.ac.cn/archives/5887

@online{kexuefm-5887,
title={变分自编码器（四）：一步到位的聚类方案},
author={苏剑林},
year={2018},
month={Sep},
url={\url{https://spaces.ac.cn/archives/5887}},
}