机器学习 第七次作业

题目 1: 列举 3 种自编码器中对隐变量的约束，写出它们对应的损失函数。

1. 线性自编码器 PCA
   对隐变量的约束：维数约束（原向量的特征维数要大于隐变量向量的特征维数）
   损失函数：$\frac{1}{n}\sum_{i=1}^{n}\left||x_i-\hat{x}_i|\right| = \left||XX^T - W^TW|\right|^2 + C$
2. 稀疏自编码器
   对隐变量的约束：目的是将特征向量经过变换得到一个稀疏向量，即尽可能地使编码后的特征向量不为零的维数尽可能少
   损失函数：$\min_{D,U}\left||X-DU|\right|^2 + \lambda\left||U|\right|_1$，其中$X$为原始输入，$D$为稀疏编码矩阵，$U$是得到的稀疏自编码矩阵。
3. 概率模型的稀疏自编码器
   对隐变量的约束：原始变量和隐变量需要服从一定的概率分布$p_{encoder}(h|x) = p_{model}(h|x), p_{decoder}(x|h) = p_{model}(x|h)$
   损失函数：$-\log\left(p_{model}(h)\right) = \sum_i\left(\lambda|h_i| - \log\frac{\lambda}{2}\right) = \Omega(h) + const$

题目 2: 假设训练数据的集合为D， 编码器在给定输入x 其隐变量z的分布为$q(z|x)$，解码器 在给定隐变量 z 其对应的x 的分布为$p(x|z)$，模型的隐变量满足$p(z)$。请从编码器和解码器对应的(x, z)联合分布的 KL 散度出发，推导变分自编码器的损失函数。下图是在实现 VAE 时常用的重抽样变换的示意图，解释 VAE 训练为什么需要做这样的变换，具体做了什么样的改变？

- 设输入x满足分布$q(x)$，则根据链式法则，编码器和解码器对应的(x, z)联合分布为$E(x,z) = q(z|x)q(x); D(x,z) = p(x|z)p(z)$，两者的 KL 散度（也就是损失函数）为

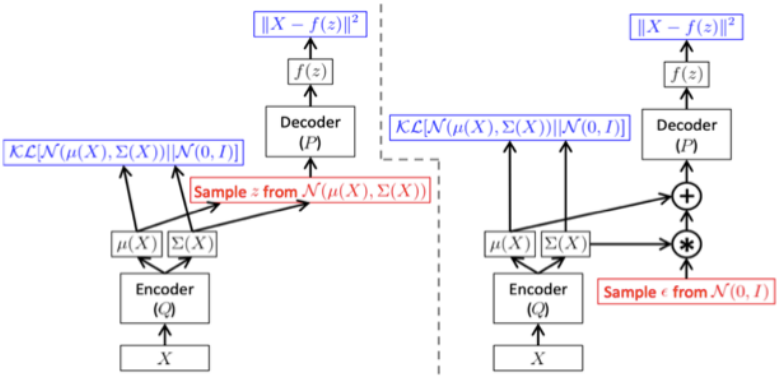
$$\begin{aligned}&\mathrm{KL[E(x)||D(x,z)]} = \mathbb{E}_{x\sim q(x),z\sim q(Z|x)}\log\left(\frac{E(x,z)}{D(x,z)}\right)\\&= \mathbb{E}_{x\sim q(x),z\sim q(Z|x)}\log\left(\frac{q(z|x)q(x)}{p(x|z)p(z)}\right)\\&= -\mathbb{E}_{x\sim q(x),z\sim q(Z|x)}[\log \mathrm{p(x|z)} + \log \mathrm{p(z)} - \log q(z|x)] + C\end{aligned}$$

- 重抽样变换的原因：

重抽样变换示意图中左侧所表示的神经网络，其前向传播能正常进行，当输出是根据分布函数$Q(z|X)$采样的大量X和z平均的结果时，结果也会是正确的。但是当我们对某一从$Q(z|X)$中对z采样的层进行误差反向传播时，我们进行的就将是非连续的操作，此时梯度将不存在。因为这个问题，所以进行了重抽样变换。

- 具体做了什么改变：

将抽样移到输入层。在给定$Q(z|X)$的均值$\mu(X)$和方差$\Sigma(X)$情况下，可以首先抽取$\epsilon \sim N(0,I)$，再计算$z=\mu(X)+\Sigma^{\frac{1}{2}}(X)*\epsilon$来从$N(\mu(X),\Sigma(X))$抽样。这种情况下，给定$X$和$\epsilon$之后操作将变成连续的了，就解决了梯度不存在的问题。



题目 3: 写出生成生成对抗模型（GAN）的损失函数，指明每个变量的意义。列出训练 GAN 时常出现的问题

损失函数为$\min_{\tilde{w}}\max_{w}\mathbb{E}_{y\sim P_{real}}F\big(D(y;w)\big)-\mathbb{E}_{\eta\sim P_{simple-distribution}}\tilde{F}(D(G(\eta;\tilde{w});w))$
变量意义：
$\mathbb{E}(*)$表示概率分布的期望值
y表示输入的原始数据
P_{real}表示y的真实分布
η表示噪声
$P_{simple-distribution}$表示噪声服从的某个简单分布。
$F,\tilde{F},D,G$均为函数。F可取log函数。D是鉴别器使用的函数，G是生成器的函数。

训练时常出现的问题：
1. 梯度消失
2. 震荡、不收敛
3. 模式塌缩
4. 高分辨率模型训练速度慢

题目 4: 下图是无监督判别式学习 SimCLR 模型的示意图。依据示意图，SimCLR 的损失函数并简述 SimCLR 模型是如何实现、训练的。

损失函数：一个正样本对(i,j)的损失函数如下。其中$1_{[k\neq i]}\in\{0,1\}$是一个指示函数，其值等于 1 当且仅当$k\neq i$，而τ表示温度参数。

$$\ell_{i,j} = -\log \frac{e^{\frac{sim(z_i,z_j)}{\tau}}}{\sum_{k=1}^{2N} 1_{[k\neq i]} e^{\frac{sim(z_i,z_j)}{\tau}}}$$

最终的损失函数则是所有正样本对的损失函数的总和，如下：

$$\mathcal{L} = \frac{1}{2N}\sum_{k=1}^{N}[\ell_{2k-1,2k} + \ell_{2k,2k-1}]$$

SimCLR 是如何实现、如何训练的：

训练过程概述：SimCLR 通过最大化对于不同参数视图下的相同数据样本的共同认同度（共识）来进行表示学习。这个过程是通过在隐空间内基于对比的损失函数进行的。这个框架由以下部分组成：

- 随机的数据增强模块 对任意给定的数据样本进行转换，基于同一数据样本生成两个相关的数据视图，用$\tilde{x}_i, \tilde{x}_j$表示，我们将其视作一个正样本对。有三种简单的数据增强方式。
- 神经网络基编码器$f(\cdot)$ 从数据增强样本中提取出表示向量。
- 另一个小的神经网络 预测头$g(\cdot)$ 其作用是将向量表示映射到进行基于对比的损失函数的计算的那个隐空间中。
- 基于对比的损失函数 用于基于对比的预测任务。

根据示意图，两个独立的数据增强算子将从同一数据增强操作簇中采样，亦即，$t\sim\mathcal{T}, t'\sim\mathcal{T}$。之后便使用这两个数据样本得到两个相关的数据视图。然后利用基于对比的损失函数对基编码器神经网络$f(\cdot)$和预测神经网络$g(\cdot)$进行训练，使得它们达成对数据的最大共识。训练完成之后，我们将舍弃预测神经网络$g(\cdot)$而使用编码器$f(\cdot)$和表示向量h进行下游的任务。