变分自编码器

分析VAE的简单实现

- [ ] $encoder$和$decoder$ 仍然是线性层和激活函数层的组合，和传统自编码器一致：

```
self.encoder = nn.Sequential(
         nn.Linear(image_size, h_dim),
         nn.LeakyReLU(0.2),
         nn.Linear(h_dim, z_dim*2))

 self.decoder = nn.Sequential(
         nn.Linear(z_dim, h_dim),
         nn.ReLU(),
         nn.Linear(h_dim, image_size),
         nn.Sigmoid())
```

- [ ] 整体结构如下：

```
def forward(self, x):
        h = self.encoder(x)
        mu, log_var = torch.chunk(h, 2, dim=1)
        z = self.reparametrize(mu, log_var)
        out = self.decoder(z)
        return out, mu, log_var
```

可以看出，VAE相较传统结构增加的就是$reparametrize$ 部分，VAE 之所以能起效就在于此。接下来查看$reparametrize$ 的作用。

- [ ] $reparametrize$函数，顾名思义，是对$encoder$ 得出的信息进行重整/采样：

```
def reparametrize(self, mu, log_var):
        eps = to_var(torch.randn(mu.size())
        z = mu + eps * torch.exp(log_var/2)
        return z
```

mu和log_var都是$batch_size * z_dim$ 大小的$tensor$ ，于是$decoder$的输入z 的大小也是相同的。现在的VAE的输入z 相比以往多了一步从正态分布采样的操作。

(从$N(\mu,\ \sigma^2)$中采样一个z,相当于从$N(0,I)$中采样一个ϵ，然后让$z=\mu+\epsilon*\sigma$)

以上都是给一个直观的印象——VAE 是什么，接下来分析为什么——为什么VAE 能起效。

为何起效

- [ ] z 作为$decoder$ 的输入，人为加了正态分布的噪声，这就比传统做法更鲁棒——“强迫”$decoder$获得去噪能力。重构误差和噪声两者在博弈与对抗。
- [ ] $P(Z|X)$是给定输入数据X情况下Z的后验分布（大写字母Z是随机变量，小写字母z是随机变量的某个具体取值）。VAE 假设此后验分布为正态分布，并用$encoder$这个神经网络去确定正态需要的两个参数——均值和方差。而神经网络最擅长的就是拟合复杂的函数/分布，这是非神经网络方法做不到的。在上述VAE 实现中用不太深的MLP就能起到效果。
- [ ] 理想的$P(Z|X)$是标准正态分布$N(0,I)$，因为在此条件下$P(Z)$也会是$N(0,I)$。而$P(Z|X)$是可以用$encoder$来逼近$N(0,I)$的。

作为桥梁的误差函数

- [ ] 简记已知事实如下：

- 先验知识：$P(Z)$可以用来生成数据，且$P(Z)$为$N(0,I)$时，均值为0向量，生成的数据既有差异也能保证“归一化”。
- $P(Z|X)$为$N(0,I)$时，可得$P(Z)$也为$N(0,I)$。

$$P(Z)=\sum_X P(Z|X)P(X)=\sum_X N(0,I)P(X)=N(0,I)\sum_X P(X)=N(0,I)$$

- [ ] 问题转化为：如何使$P(Z|X)$为$N(0,I)$呢?

- [ ] 上文已提到，$encoder$用来确定$P(Z|X)$，那就用KL散度作为当前的$P(Z|X)$和我们预设的$N(0,I)$ 的差异度量。

$$KL(P(Z|X)||N(0,I))=-\frac{1}{2}(-log\sigma^2+\mu^2+\sigma^2+1)$$

显然上面的式子可以用梯度下降去解。当其越接近0，$P(Z|X)$也就越接近$N(0,I)$。这也就是$encoder$部分的误差函数，我们可将其称为**编码误差**。

- [ ] 接下来考虑$decoder$部分，理所当然，最终的误差函数要包括**重构误差**。也就是衡量VAE的输入X和输出$\hat{X}$的差异。如果输入输出相差太大，说明重构失败，模型崩坏。如果输入输出相差无几，VAE作为目的为生成数据的$generative\ model$ 又失去意义。我们的目标是生成的数据既有多样性又抓住输入的规律。
- [ ] 所以最终的误差函数是：

$$总误差 = 重构误差 + 编码误差 * 正则系数$$

其实这也可以看成，统计机器学习中关于经验风险和结构风险的理论在神经网络领域的对应。

深层些的分析

- [ ] $encoder$的实质是$P(Z|X)$，$decoder$则是$P(X|Z)$，人人争说贝叶斯，而这就是贝叶斯。VAE在生成模型里未必是效果最好的，但它将概率图模型和深度学习统一在同个模型中。
- [ ] 换言之，在VAE中，我们将隐藏编码对应的节点看成是随机变量，其它节点还是作为普通神经元。