AE & VAE

目录

AE

一、 理论学习

没有找到自编码器最初的提出，所以根据邱锡鹏《神经网络和深度学习》P214-P221 来学习了自编码器。

自编码器大概可以总结为两个基本的神经网络的结合。第一个神经网络是编码器，可以理解为一个压缩或数据降维的过程。第二个神经网络是解码器，原来的很多个 feature，经过第一个神经网络压缩成比较少个 feature 来代表原来的数据，再通过解码器解压之后恢复成原来的维度。它的两个部分可以表示为：

编码器（Encoder）$f: \mathbb{R}^D \rightarrow \mathbb{R}^M$

解码器（Decoder）g: $\mathbb{R}^M \rightarrow \mathbb{R}^D$

自编码器的学习目标是最小化重构错误：

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \| \boldsymbol{x}^{(n)} - \mathbf{g}(f(\boldsymbol{x}^{(n)})) \|^2 \\ &= \sum_{n=1}^{N} \| \boldsymbol{x}^{(n)} - f \circ \mathbf{g}(\boldsymbol{x}^{(n)}) \|^2\end{aligned}$$

二、 网络构建

一般自编码器的结构其实非常简单，就是两个基本的神经网络。而这两个神经网络可以采用一般神经网络的结构。所以接下来将记录两种自编码器的构建，一种是全连接自编码器，另一种是卷积自编码器。

1．卷积自编码器

卷积自编码器的构造难点在于，对于复杂一点的卷积自编码器，两个神经网络并不用完全对称，需要注意各层的输入输出维度，最后将维度映射到（28，28，1）即可。

1) 对反卷积和上采样的理解

很多资料对这两个概念介绍地非常模糊，导致我一开始认为反卷积是针对卷积层的逆映射操作，上采样是针对池化层的逆映射操作，卷积和反卷积，池化和上采样要一一对应构建。

而其实，反卷积和上采样都是使低维度的数据恢复高维度的操作。

“反卷积”并不是完全的“逆映射”操作，使用的还是卷积操作，用转置卷积核权重和合适的步长、填充大小来实现特征图的**扩大**（或者单纯地继续提取特征，不扩大特征图）。而“上采样”的目的就是使用插值方法或填充操作**扩大特征图**，使低维度的数据恢复高维。

两者有一定的关系。一些反卷积的操作可以认为是上采样操作，其目的就是扩大特征图。对于更多的反卷积操作，其也是卷积，其参数可学习，能够更有效地重建原始输入。

2) 代码实现

①编码器

一共使用了 4 次卷积，三次池化，维度从（28，28，1）到（4，4，8）。也就说输入的是 28*28*1 的图片，输出是 4*4*8 的 feature maps

②解码器

一共使用了 5 次卷积，3 次上采样，维度从（4*4*8）到（28，28，1）。

③数据处理

数据处理时将数据归一化到[0,1]（因为激活函数使用的是 ReLU 函数）。然后加噪声来观察自编码器降噪的效果。

2．全连接自编码器

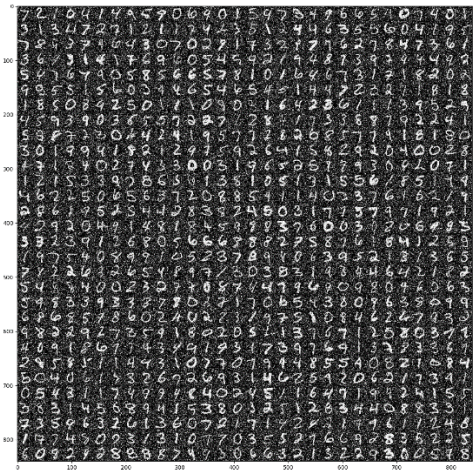
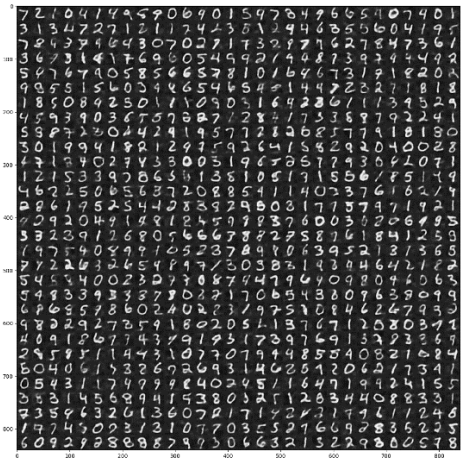
全连接自编码器实现非常简单，因为都是全连接层，选择合适的隐藏层节点即可。另外输入输出的维度转换需要注意。除了编码器和解码器的部分以及损失函数不一样，其他和 CAE 的实现是一模一样的。

3．两者对数据降噪的效果对比

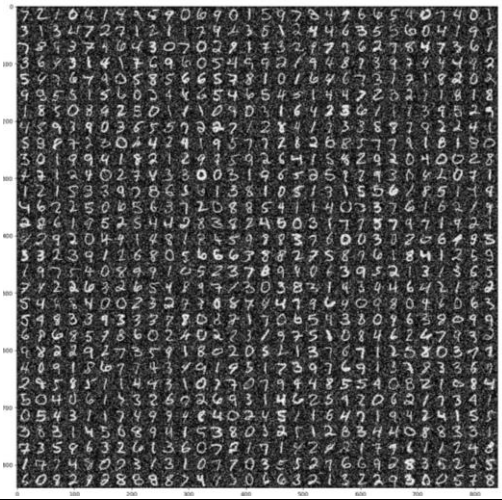
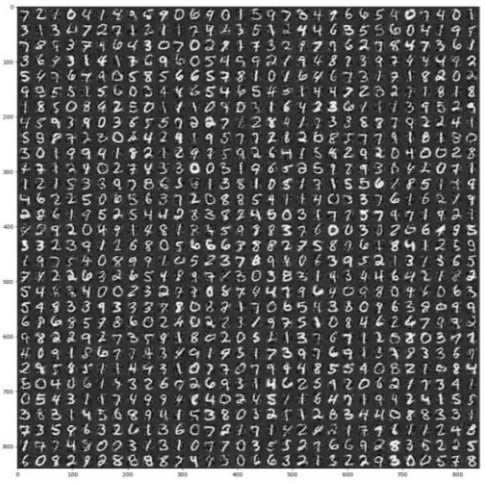
训练轮数：20

1) 对于 digits 数据

卷积

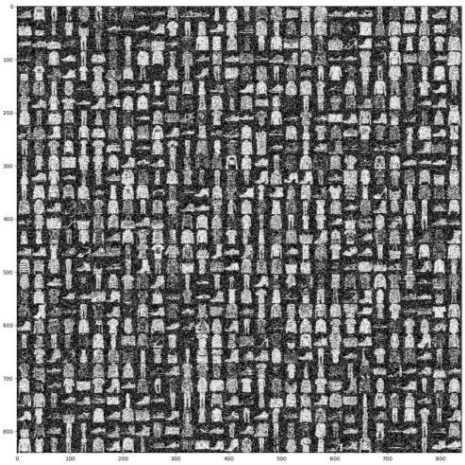
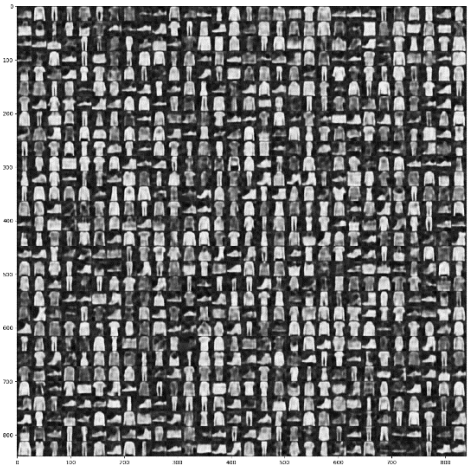
含噪图片	降噪图片
	

全连接

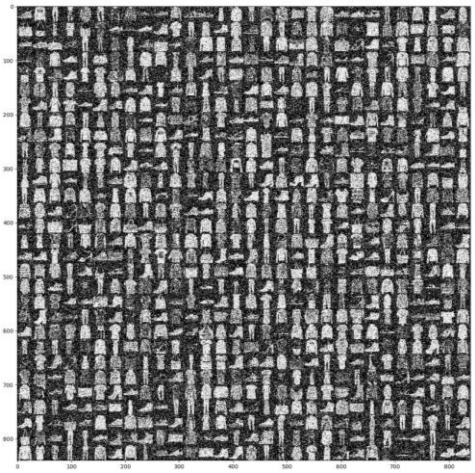
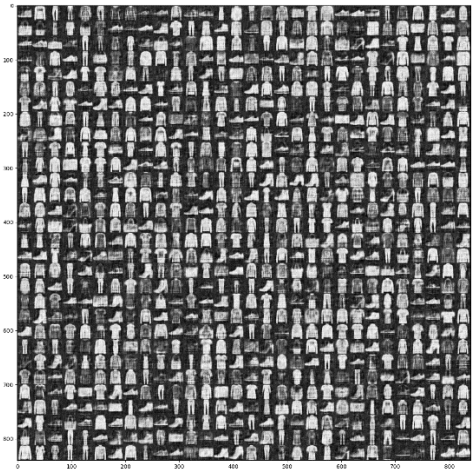
含噪图片	降噪图片
	

2) 对于 fashion 数据

卷积

含噪图片	降噪图片
	

全连接

含噪图片	降噪图片
	

因为没有找到合适的可以衡量效果的指标，所以只能采用主观判断。对于 digits 数据，卷积自编码器效果更好是比较明显的，图片上的一些痕迹都处理得比较干净，而全连接自编码器的生成结果还存在着较多的噪声。对于 fashion 数据，两者在肉眼上没有较大的差距，甚至全连接自编码器效果更好一点。全连接自编码器的优势在于神经网络非常简单，训练非常快。两者效果因数据集而异。

VAE

一、 理论学习

1．理论解读

在阅读学习的过程中对 VAE 的理论基础按照我自己的思路做了一遍梳理。

首先，“变分”是指一种近似推断方法，用于处理复杂概率模型下的推断问题。变分自编码器是一种概率生成模型，一般用于密度估计。

《神经网络与深度学习》中这段话将变分自编码器的目标讲得比较明确：

> 在机器学习中，密度估计是一类无监督学习问题. 比如在手写体数字图像的密度估计问题中，我们将图像表示为一个随机向量 $\boldsymbol{X}$，其中每一维都表示一个像素值. 假设手写体数字图像都服从一个未知的分布 $p_r(\boldsymbol{x})$，希望通过一些观测样本来估计其分布. 但是，手写体数字图像中不同像素之间存在复杂的依赖关系（比如相邻像素的颜色一般是相似的），很难用一个明确的图模型来描述其依赖关系，所以直接建模 $p_r(\boldsymbol{x})$ 比较困难. 因此，我们通常通过引入隐变量 $\boldsymbol{z}$ 来简化模型，这样密度估计问题可以转换为估计变量 $(\boldsymbol{x}, \boldsymbol{z})$ 的两个局部条件概率 $p_\theta(\boldsymbol{z})$ 和 $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. 一般为了简化模型，假设隐变量 $\boldsymbol{z}$ 的先验分布为标准高斯分布 $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. 隐变量 $\boldsymbol{z}$ 的每一维之间都是独立的. 在这个假设下，先验分布 $p(\boldsymbol{z}; \theta)$ 中没有参数. 因此，密度估计的重点是估计条件分布 $p(\boldsymbol{x}|\boldsymbol{z}; \theta)$.

变分自编码器实际上是建模含隐变量的生成模型。主要利用 EM 算法来进行密度估计条件分布$p(x|z;\theta)$以及后验分布$p(z|x;\theta)$，而因为这两个分布比较复杂，在变分自编码器中使用神经网络来建模。

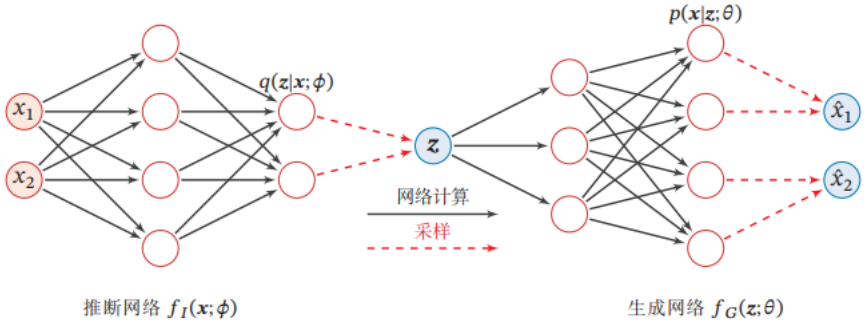
1） VAE 的目标

变分自编码器是含隐变量的生成模型，所以最初的目标是使用最大似然来估计联合概率$p(x,z;\theta)$。它的联合概率密度函数为

$$p(x,z;\theta) = p(x \mid z;\theta)p(z;\theta) \tag{1}$$

而最终 VAE 要生成估计的样本 x，是根据隐变量来生成的。所以它的网络结构分为推断网络和生成网络。推断网络的最终目的是根据概率分布$p(z \mid x;\theta)$来采样

z。而生成网络的最终目的是根据隐变量 z 来生成 x（有点类似于 CGAN 的生成网络）。其网络结构为：



2） 推断网络

推断网络的输入为 x，输出为变分分布$q(z|x;\phi)$（即条件分布$p(z \mid x;\theta)$的近似估计）**的参数ϕ***。然后根据变分分布来采样隐变量 z（假设 z 的先验分布就是一些常见分布，例如高斯分布）在这里，为了估计$p(z \mid x;\theta)$，使用 KL 距离作为估计指标。

根据一些列数学推导，KL 距离可以表示为

$$\mathrm{KL}\big(q(\boldsymbol{z};\phi),p(\boldsymbol{z} \mid \boldsymbol{x};\theta)\big) = \log p(\boldsymbol{x};\theta) - ELBO(q,\boldsymbol{x};\theta,\phi) \qquad (2)$$

在公式中，前一项和参数ϕ无关。$ELBO(q,\boldsymbol{x};\theta,\phi)$的公式为：

$$ELBO(q,x;\theta,\phi) = \sum_z q(z;\phi)\log\frac{p(x,z;\theta)}{q(z;\phi)} = \mathbb{E}_{z\sim q(z;\phi)}[\log\frac{p(x,z;\theta)}{q(z;\phi)}]$$

所以该推断网络的优化目标是最小化 KL 距离，也就是**最大化$\boldsymbol{ELBO(q,x;\theta,\phi)}$**，即推断网络的目标转换为寻找一组网络参数$\phi$*（权重、偏置等）使得$ELBO(q,x;\theta,\phi)$最大。这可以看作是 EM 算法中的 E 步。

3） 生成网络

生成网络输入为采样的 z，输出为概率分布$p(x,z;\theta)$的**参数$\boldsymbol{\theta}$***。而根据公式(1)，首先$p(z;\theta)$是已知的分布（《神经网络和深度学习》中假设隐变量$\boldsymbol{z}$的先验分布为各向同性的标准高斯分布，隐变量$\boldsymbol{z}$的每一维之间都是独立的），所以目标转换为就是计算$p(x \mid z;\theta)$。

在 VAE 中，采用神经网络来建模$p(x \mid z;\theta)$。假设$p(x \mid z;\theta)$服从常见的分布（二值向量就是伯努利分布，高维向量就是对角化协方差高斯分布），然后用神

经网络来估计这些分布需要的参数θ。还是采用最大似然估计来估计$p(x \mid z;\theta)$。根据一系列的数学推导和全概率定理：

$$log\ p(x|z;\theta) = \int_z q(z;\phi)\,[log\ p(x|z;\theta)$$

$$= \mathbb{E}_{z\sim q(z;\phi)}[log\ \ p(x \mid z;\theta)]$$

$$= ELBO(q,x;\theta,\phi) - KL(q(\boldsymbol{z};\phi),p(\boldsymbol{z};\theta))$$

而根据$p(\boldsymbol{z};\theta)$已知，在生成网络中$\boldsymbol{\phi}$已知，所以 KL 距离已知。所以要最大似然估计$p(x \mid z;\theta)$相当于**最大化$\boldsymbol{ELBO(q,x;\theta,\phi)}$**。所以生成网络的目标转换为寻找一组网络参数$\theta*$（权重、偏置等）使得$ELBO(q,x;\theta,\phi)$最大。这可以看作是 EM 算法中的 M 步。

4) 目标函数

综合 2)和 3)，总结得出 VAE 的总目标函数为

$$\max_{\theta,\phi} ELBO(q,\boldsymbol{x};\theta,\phi) = \max_{\theta,\phi} \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\phi})}\left[\log\frac{p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})p(\boldsymbol{z};\boldsymbol{\theta})}{q(\boldsymbol{z};\boldsymbol{\phi})}\right]$$

$$= \max_{\theta,\phi} \mathbb{E}_{z\sim q(z|x;\phi)}[\log p(x \mid z;\theta)] - \mathrm{KL}\big(q(z \mid x;\phi),p(z;\theta)\big) \quad (3)$$

对于公式(3)中的第二项，即 KL 散度一般可以直接计算。对于第一项，可以通过采样的方式近似计算。对于生成网络是可以直接近似估计，但对于推断网络，存在一点问题。随意采用再参数化和梯度估计的方法来解决。由于实现的时候使用了再参数化，所以这里也只记录再参数化方法。

5) 再参数化

公式（2）中的第一项期望$\mathbb{E}_{z\sim q(z|x;\phi)}[\log\ p(x \mid z;\theta)]$可以通过采样的方式近似计算。对于每个样本$\boldsymbol{x}$，根据$q(z \mid x;\phi)$采集 M 个$\boldsymbol{z^m}$，有：

$$\mathbb{E}_{z\sim q(z|x;\phi)}[\log\ p(x \mid z;\theta)] \approx \frac{1}{M}\sum_{m=1}^{M} \log\ p(x \mid z^{(m)};\theta)$$

正是由于“采样”的操作，所以$\boldsymbol{z}$和$\boldsymbol{\phi}$之间不是确定性关系。

→***<u>确定性关系</u>***：在概率论中，如果两个变量之间的关系是确定性的，意味着当一个变量的值发生改变时，另一个变量的值也会相应地发生确定性的改变，即

它们之间存在一个确定的函数关系。在 VAE 中，$\boldsymbol{z}$是通过对某个固定分布（通常是高斯分布）进行采样得到的，而采样过程是通过变分分布$q(\boldsymbol{z}|\boldsymbol{x};\ \boldsymbol{\phi})$的参数$\boldsymbol{\phi}$进行控制的。当我们改变参数$\boldsymbol{\phi}$时，变分分布发生了变化，从而会影响采样过程，进而影响$\boldsymbol{z}$的取值。但$\boldsymbol{z}$和$\boldsymbol{\phi}$之间的这种影响并不是以确定性的、可预测的方式发生的。

因此，由于采样得到的$\boldsymbol{z}$与参数$\boldsymbol{\phi}$之间没有确定性的函数关系，**无法直接计算$\boldsymbol{z}$关于$\boldsymbol{\phi}$的导数**。这时，就可以通过再参数化方法来将$\boldsymbol{z}$和$\boldsymbol{\phi}$ 之间随机性的采样关系转变为确定性函数关系。

引入一个分布为$p(\epsilon)\sim \mathcal{N}(\boldsymbol{0},\ \boldsymbol{I})$的随机变量$\epsilon$，期望$\mathbb{E}_{z\sim q(z|x;\phi)}[\log\ p(x\mid z;\theta)]$可以重写为：

$$\mathbb{E}_{z\sim q(z|x;\phi)}[\log\ p(x\mid z;\theta)]=\mathbb{E}_{\varepsilon\sim p(\varepsilon)}[\log\ p(x\mid g(\phi,\varepsilon);\theta)]$$

假设$q(\boldsymbol{z}|\boldsymbol{x};\ \boldsymbol{\phi})$为正态分布$N(\boldsymbol{\mu}_I,\boldsymbol{\sigma}_I^2\boldsymbol{I})$，其中$\{\mu_I,\sigma_I\}$是推断网络的输出，依赖于参数$\boldsymbol{\phi}$，可以通过下面方式来再参数化：

$$z=\mu_I+\sigma_I\odot\varepsilon \tag{4}$$

这样 $\boldsymbol{z}$ 和参数 $\boldsymbol{\phi}$ 的关系从采样关系变为确定性关系，从而可以求$\boldsymbol{z}$关于$\boldsymbol{\phi}$的导数。

6） 目标函数的化简和近似

给定一个数据集$\mathcal{D}=\{x^{(n)}\}_{n=1}^N$，对于每个样本$\boldsymbol{x}^{(n)}$，随机采样 M 个变量$\epsilon^{(n,m)}$，并计算$z^{(n,m)}$，则变分自编码器的目标函数近似为：

$$\mathcal{J}(\phi,\theta\mid\mathcal{D})=\sum_{n=1}^{N}(\frac{1}{M}\sum_{m=1}^{M}\log\ p(x^{(n)}\mid z^{(n,m)};\theta)-\mathrm{KL}(q(z\mid x^{(n)};\phi),\mathcal{N}(z;0,I)))$$

采用随机梯度方法，每次从数据集中采集一个样本$\boldsymbol{x}$和一个对应的随机变量ϵ，并假设$p(x\mid z;\theta)\sim\mathcal{N}(\boldsymbol{x}\mid\boldsymbol{\mu}_G,\lambda\boldsymbol{I})$，其中$\boldsymbol{\mu}_G$是生成网络的输出，$\lambda$为超参，则目标函数进一步化简为

$$J(\phi,\theta\mid\boldsymbol{x})=-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}_G\|^2-\lambda\mathrm{KL}\big(\mathcal{N}(\boldsymbol{\mu}_I,\boldsymbol{\sigma}_I),\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\big) \tag{5}$$

第一项可以近似看作输入 $\boldsymbol{x}$ 的**重构正确性**，第二项可以看作**正则化项**。

7) 对目标函数的另一种解读

VAE 的原始目标函数为：

$$\max_{\theta,\phi} ELBO(q,\boldsymbol{x};\theta,\phi)$$

上面的记录主要是参考《神经网络和深度学习》，将 VAE 的两种网络分开来讨论。下面将把 VAE 看成是一个网络来推导出它的目标函数。

这里参考机器学习方法–优雅的模型（一）：变分自编码器（VAE） - 知乎 (zhihu.com)。

VAE 最终的输出是 x，也就是要估计分布$\boldsymbol{p_\theta(x)}$。仍然是使用 MLE 方法，那么就是要最大化$\boldsymbol{log\ p_\theta(x)}$。根据公式(2)

$$\log\ p(x;\theta) = ELBO(q,x;\theta,\phi) + \mathrm{KL}(q(z;\phi),p(z\mid x;\theta))$$

也就是说：

$$ELBO(q,x;\theta,\phi) = log\ p(x;\theta) - KL(q(z;\phi),p(z\mid x;\theta)) \quad (3)$$

再结合推断网络和生成网络各自的目标，推断网络需要最小化 KL 距离，生成网络需要最大化$log\ p(x;\theta)$。

所以，根据公式(3)，只需要优化$ELBO(q,x;\theta,\phi)$，就可以最大化$log\ p(x;\theta)$，并且最小化$\mathrm{KL}(q(z;\phi),p(z\mid x;\theta))$。所以最终的目标函数就是：

$$\max_{\theta,\phi} ELBO(q,\boldsymbol{x};\theta,\phi)$$

2．创新点

1) 变分推断和神经网络的结合来拟合很复杂的分布

2) 隐变量的引入和条件概率的使用

3) 再参数化技巧的使用

虽然 VAE 名字中有“自编码器”，但是和自编码器还是很不一样的。因为它的输出是概率分布而不是所谓的编码。相同的点在于两个网络的构建和连接。

二、 网络构建

1．代码实现

代码参考：

keras-io/examples/generative/vae.py at master · keras-team/keras-io · GitHub

根据官方代码重构了 VAE 网络。

1) 推断网络和生成网络的结构

推断网络和生成网络内部采用了卷积网络，具体都是采用 keras 搭建完成。这里只记录两个网络的输入输出。

①推断网络

根据上文推断网络中所记录的输入输出项，在代码中这样设计：

```
# 推断网络
def build_encoder(self):
    # 输入
    input = Input(shape=self.img_shape)
    x = model(input)
    # 输出
    # z|x 的分布参数
    z_mean = Dense(self.latent_dim, name="z_mean")(x)
    z_log_var = Dense(self.latent_dim, name="z_log_var")(x)
    # z 的采样
    z = self.Sampling([z_mean, z_log_var])

    return Model(input, [z_mean, z_log_var, z], name='encoder')
```

②生成网络

```
# 生成网络
def build_decoder(self):
    # 输入 采样的 z
    input = Input(shape=(self.latent_dim,))
    # 输出 x|z 的分布参数
    output = model(input)

    return Model(input, output, name='decoder')
```

2) 再参数化技巧的应用

主要是用于采样 z，根据公式(4)，z 的采样过程这样设计：

```
# 采样 再参数化采样
def Sampling(self, inputs):
    z_mean, z_log_var = inputs
    batch = tf.shape(z_mean)[0]
    dim = tf.shape(z_mean)[1]
    epsilon = tf.random.normal(shape=(batch, dim))
    return z_mean + tf.exp(0.5 * z_log_var) * epsilon
```

3) 损失函数/目标函数的设计

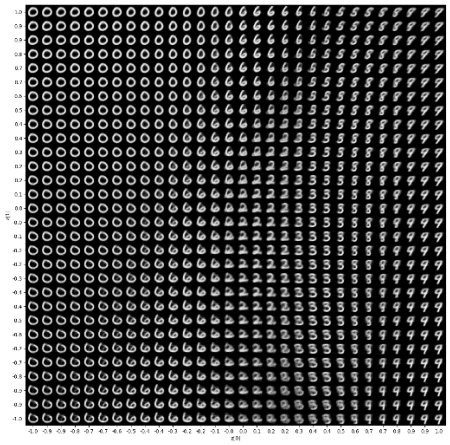
VAE 和其他一般的网络结构不一样，不能直接使用现成的损失函数。根据公式(5)，损失函数这样设计：

```
z_mean, z_log_var, z = self.encoder(data)
reconstruction = self.decoder(z)
# 重构正确性
reconstruction_loss = tf.reduce_mean(
    tf.reduce_sum(
        binary_crossentropy(data, reconstruction), axis=(1, 2)
    )
)
# 负的 KL 距离
kl_loss = -0.5 * (1 + z_log_var - tf.square(z_mean) - tf.exp(z_log_var))
kl_loss = tf.reduce_mean(tf.reduce_sum(kl_loss, axis=1))
# ELOB
total_loss = reconstruction_loss + kl_loss
```

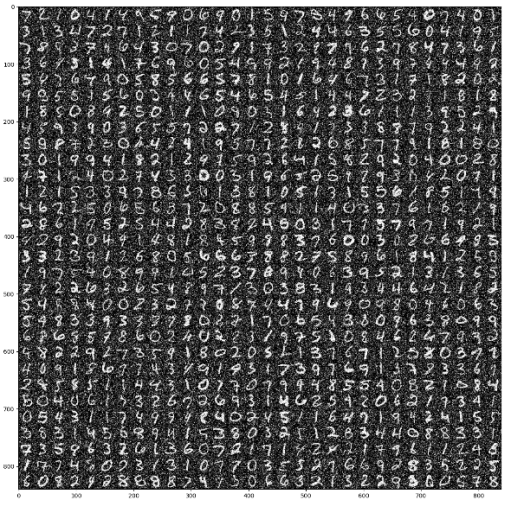
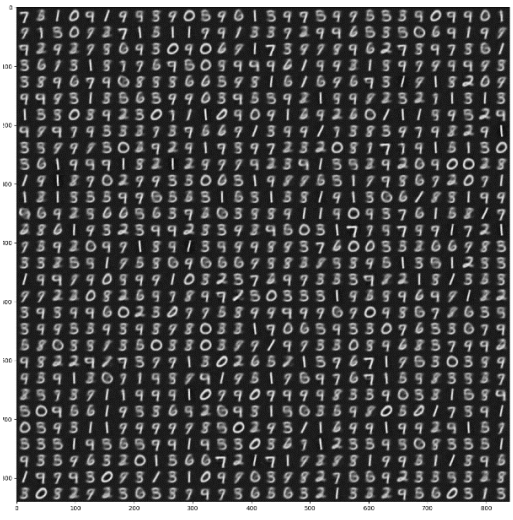
2．结果展示

训练轮数：15

1) 图片生成效果



2) 降噪效果

含噪图片	降噪图片
	

可以看到 VAE 的降噪效果是非常不错的，噪声处理得比较干净。生成的图片效果也比较好

AE & VAE 实验总结

AE 的实现比较简单，只要选择合适的神经网络搭建成编码器和解码器即可。VAE 的实现难点在于对 VAE 中理论的理解，它涉及到大量的条件概率和数学推导，理解起来还是比较晦涩的。但是在实现上并不是很复杂。

参考文献

VAE:

1. 邱锡鹏《神经网络与深度学习》P310-P317
2. Diederik P. Kingma *Auto-Encoding Variational Bayes*
3. 机器学习方法–优雅的模型（一）:变分自编码器（VAE） - 知乎 (zhihu.com)