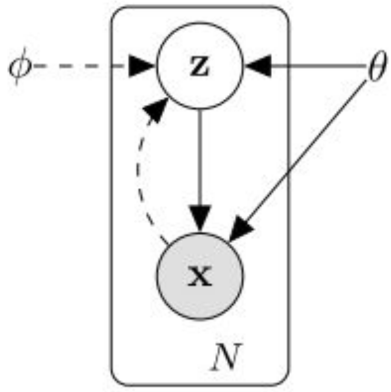
VAEs简介

变分自编码器（Variational auto-encoder，VAE）是一类重要的生成模型（generative model），它于2013年由Diederik P.Kingma和Max Welling提出[1]。2016年Carl Doersch写了一篇VAEs的tutorial[2]，对VAEs做了更详细的介绍，比文献[1]更易懂。这篇读书笔记基于文献[1]。

除了VAEs，还有一类重要的生成模型GANs（对GANs感兴趣可以去我的微信公众号看介绍文章：学术兴趣小组）。

我们来看一下VAE是怎样设计的。



上图是VAE的图模型。我们能观测到的数据是 x，而 x 由隐变量 z 产生，由 $\mathrm{z} \to \mathrm{x}$ 是生成模型 $p_\theta(\mathrm{x}|\mathrm{z})$，从自编码器（auto- encoder）的角度来看，就是解码器；而由 $\mathrm{x} \to \mathrm{z}$ 是识别模型（recognition model） $q_\phi(\mathrm{z}|\mathrm{x})$，类似于自编码器的编码器。

VAEs现在广泛地用于生成图像，当生成模型 $p_\theta(\mathrm{x}|\mathrm{z})$ 训练好了以后，我们就可以用它来生成图像了。与GANs不同的是，我们是知道图像的密度函数（PDF）的（或者说，是我们设定的），而GANs我们并不知道图像的分布。

VAEs模型的理论推导

以下的推导参考了文献[1]和[3]，文献[3]是变分推理的课件。

首先，假定所有的数据都是独立同分布的（i.i.d），两个观测不会相互影响。我们要对生成模型 $p_\theta(\mathrm{x}|\mathrm{z})$ 做参数估计，利用对数最大似然法，就是要最大化下面的对数似然函数：

$$\log p_\theta(\mathrm{x}^{(1)}, \mathrm{x}^{(2)}, \cdots, \mathrm{x}^{(N)}) = \sum_{i=1}^{N} \log p_\theta(\mathrm{x}^{(i)})$$

VAEs用识别模型 $q_\phi(\mathrm{z}|\mathrm{x}^{(i)})$ 去逼近真实的后验概率 $p_\theta(\mathrm{z}|\mathrm{x}^{(i)})$，衡量两个分布的相似程度，我们一般采用KL散度，即

$$\begin{aligned}
KL(q_\phi(\mathrm{z}|\mathrm{x}^{(i)})||p_\theta(\mathrm{z}|\mathrm{x}^{(i)})) &= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})}{p_\theta(\mathrm{z}|\mathrm{x}^{(i)})} && (1)\\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x}^{(i)}) p_\theta(\mathrm{x}^{(i)})}{p_\theta(\mathrm{z}|\mathrm{x}^{(i)}) p_\theta(\mathrm{x}^{(i)})} && (2)\\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})}{p_\theta(\mathrm{z}, \mathrm{x}^{(i)})} + \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})} \log p_\theta(\mathrm{x}^{(i)}) && (3)\\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x}^{(i)})}{p_\theta(\mathrm{z}, \mathrm{x}^{(i)})} + \log p_\theta(\mathrm{x}^{(i)}) && (4)
\end{aligned}$$

于是

$\log p_{\theta}(\text{x}^{(i)})=KL(q_{\phi}(\text{z}|\text{x}^{(i)}),p_{\theta}(\text{z}|\text{x}^{(i)}))+\mathcal{L}(\theta,\phi;\text{x}^{(i)})$

其中，

$$\begin{aligned}\mathcal{L}(\theta,\phi;\text{x}^{(i)}) &= -\mathbb{E}_{q_{\phi}(\text{z}|\text{x}^{(i)})}\log\frac{q_{\phi}(\text{z}|\text{x}^{(i)})}{p_{\theta}(\text{z},\text{x}^{(i)})} && (5)\\ &= \mathbb{E}_{q_{\phi}(\text{z}|\text{x}^{(i)})}\log p_{\theta}(\text{z},\text{x}^{(i)})-\mathbb{E}_{q_{\phi}(\text{z}|\text{x}^{(i)})}\log q_{\phi}(\text{z}|\text{x}^{(i)}) && (6)\end{aligned}$$

由于KL散度非负，当两个分布一致时（允许在一个零测集上不一致），KL散度为0。于是 $\log p_{\theta}(\text{x}^{(i)})\geq\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 。 $\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 称为对数似然函数的变分下界。

直接优化 $\log p_{\theta}(\text{x}^{(i)})$ 是不可行的，因此一般转而优化它的下界 $\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 。对应的，优化对数似然函数转化为优化 $\mathcal{L}(\theta,\phi;\text{X})=\displaystyle\sum_{i=1}^{N}\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 。

作者指出， $\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 对 ϕ 的梯度方差很大，不适于用于数值计算。为了解决这个问题，假定识别模型 $q_{\phi}(\text{z}|\text{x})$ 可以写成可微函数 $g_{\phi}(\epsilon,\text{x})$ ，其中， ϵ 为噪声， $\epsilon\sim p(\epsilon)$ 。于是， $\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 可以做如下估计（利用蒙特卡罗方法估计期望）：

$\tilde{\mathcal{L}}^{A}(\theta,\phi;\text{x}^{(i)})=\dfrac{1}{L}\displaystyle\sum_{l=1}^{L}[\log p_{\theta}(\text{x}^{(i)},\text{z}^{(i,l)})-\log q_{\phi}(\text{z}^{(i,l)}|\text{x}^{(i)})]$

其中， $\text{z}^{(i,l)}=g_{\phi}(\epsilon^{(i,l)},\text{x}^{(i)}),\quad \epsilon^{(i,l)}\sim p(\epsilon)$ 。

此外， $\mathcal{L}(\theta,\phi;\text{x}^{(i)})$ 还可以改写为

$\displaystyle \mathcal{L}(\theta,\phi;\text{x}^{(i)})=-KL(q_{\phi}(\text{z}|\text{x}^{(i)})||p_{\theta}(\text{z}))$

- $+\mathbb{E}_{q_{\phi}(\text{z}|\text{x}^{(i)})} \log p_{\theta}(\text{x}^{(i)}|\text{z})$

由此可以得到另外一个估计

$\tilde{\mathcal{L}}^{B}(\theta,\phi;\text{x}^{(i)})=-KL(q_{\phi}(\text{z}|\text{x}^{(i)})||p_{\theta}(\text{z}))+\dfrac{1}{L}\displaystyle\sum_{l=1}^{L}\log p_{\theta}(\text{x}^{(i)}|\text{z}^{(i,l)})$

其中， $\text{z}^{(i,l)}=g_{\phi}(\epsilon^{(i,l)},\text{x}^{(i)}),\quad \epsilon^{(i,l)}\sim p(\epsilon)$ 。

实际试验时，如果样本量 N 很大，我们一般采用minibatch的方法进行学习，对数似然函数的下界可以通过minibatch来估计：

$\mathcal{L}(\theta,\phi;\text{X})\simeq\tilde{\mathcal{L}}^{M}(\theta,\phi;\text{X}^{M})=\dfrac{N}{M}\displaystyle\sum_{i=1}^{M}\tilde{\mathcal{L}}(\theta,\phi;\text{x}^{(i)})$

可以看到，为了计算 $\mathcal{L}(\theta,\phi;\text{X})$ ，我们用了两层估计。当 M 较大时，内层估计可以由外层估计来完成，也就是说，取 $L=1$ 即可。实际计算中，作者取 $M=100,L=1$ 。由上述推导得到AEVB算法：

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta}, \boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

VAEs模型

上面给的AEVB算法是一个算法框架，只有给定了 $\epsilon, p_\theta(\mathrm{x}|\mathrm{z}), q_\phi(\mathrm{z}|\mathrm{x}), p_\theta(\mathrm{z})$ 分布的形式以及 $g_\phi(\epsilon, \mathrm{x})$ ，我们才能启动算法。实际应用中，作者取

$$
\begin{aligned}
p(\epsilon) &= \mathcal{N}(\epsilon; 0, \mathrm{I}) &(7)\\
q_\phi(\mathrm{z}|\mathrm{x}^{(i)}) &= \mathcal{N}(\mathrm{z}; \mu^{(i)}, \sigma^{2(i)}\mathrm{I}) &(8)\\
p_\theta(\mathrm{z}) &= \mathcal{N}(\mathrm{z}; 0, \mathrm{I}) &(9)\\
g_\phi(\epsilon^{(l)}, \mathrm{x}^{(i)}) &= \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)} &(10)
\end{aligned}
$$

而 $p_\theta(\mathrm{x}|\mathrm{z})$ 根据样本是实值还是二元数据进行选择，若样本为二元数据，则选择

$$p_\theta(x_i|\mathrm{z}) = \mathcal{B}(x_i; 1, y_i) = y_i^{x_i} \cdot (1 - y_i)^{1-x_i}, \quad i = 1, 2, \cdots, D_\mathrm{x}(D_\mathrm{x} = \dim(\mathrm{x}))$$

若样本是实值数据，则选择

$$p_\theta(\mathrm{x}^{(i)}|\mathrm{z}) = \mathcal{N}(\mathrm{x}^{(i)}; \mu'^{(i)}, \sigma'^{2(i)}\mathrm{I})$$

实验中，作者选择多层感知器（MLP）对 $p_\theta(\mathrm{x}|\mathrm{z}), q_\phi(\mathrm{z}|\mathrm{x})$ 进行拟合，具体来说，

对 $p_\theta(\mathrm{x}|\mathrm{z})$ ，参数为 $\theta = (\mu', \sigma')$ ，若样本为二元数据，则

$$
\begin{aligned}
\log p(\mathrm{x}|\mathrm{z}) &= \sum_{i=1}^{D_\mathrm{x}} x_i \log y_i + (1 - x_i) \cdot \log(1 - y_i) &(11)\\
\mathrm{y} &= \mathrm{sigmoid}(\mathrm{W}_2 \tanh(\mathrm{W}_1 \mathrm{z} + \mathrm{b}_1) + \mathrm{b}_2) &(12)
\end{aligned}
$$

若样本为实值数据，则

$$
\begin{aligned}
\mu' &= \mathrm{W}_4 \mathrm{h}' + \mathrm{b}_4 &(13)\\
\sigma' &= \mathrm{W}_5 \mathrm{h}' + \mathrm{b}_5 &(14)\\
\mathrm{h}' &= \tanh(\mathrm{W}_3 \mathrm{z} + \mathrm{b}_3) &(15)
\end{aligned}
$$

对 $q_\phi(\mathrm{z}|\mathrm{x})$ ，参数为 $\phi = (\mu, \sigma)$ ，

$$
\begin{aligned}
\mu &= \mathrm{W}_7 \mathrm{h} + \mathrm{b}_7 &(16)\\
\sigma &= \mathrm{W}_8 \mathrm{h} + \mathrm{b}_8 &(17)\\
\mathrm{h} &= \tanh(\mathrm{W}_6 \mathrm{x} + \mathrm{b}_6) &(18)
\end{aligned}
$$

根据以上假设的分布，不难计算

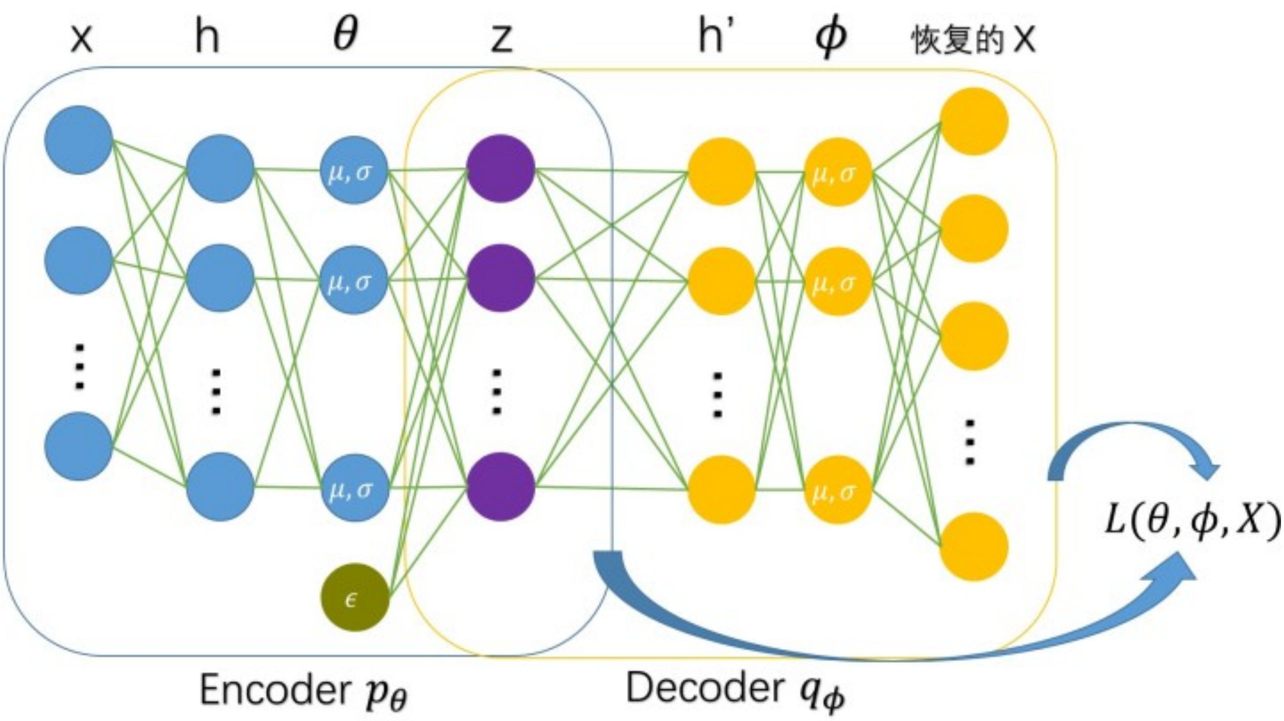
$$\mathcal{L}(\theta, \phi; \mathrm{x}^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{D_\mathrm{z}} (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\mathrm{x}^{(i)}|\mathrm{z}^{(i,l)})$$

其中， $\mathrm{z}^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)}, \quad \epsilon^{(l)} \sim p(\epsilon)$ 。

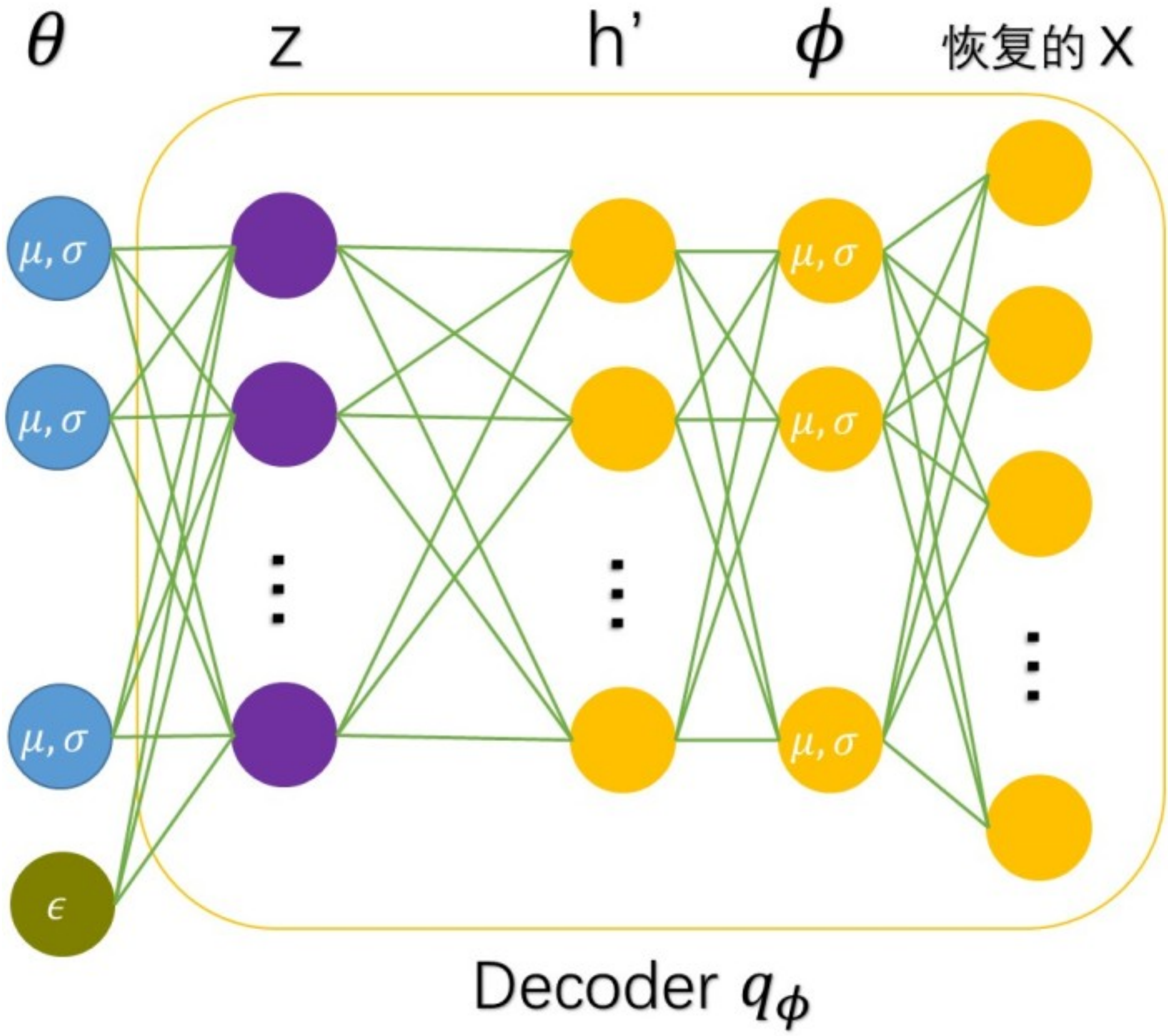
loss的推导：

$D_{KL}\left(q_\phi(z \mid x) || p_\theta(z)\right), p_\theta(z) \sim \mathrm{N}(0,1)$, 下面推导过程将 $(q_\phi(z \mid x)$ 简化为 q
$D_{KL}\left(q_\phi(z \mid x) || p_\theta(z)\right) = \int q(z) \log \frac{q(z)}{p(z)} dz = \int q(z)((\log q(z) - \log p(z))dz$
$= \int q(z)\left(\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(z-\mu)^2}{2\sigma^2}}\right) - \log\left(\frac{1}{\sqrt{2\pi}} e^{\frac{(z)^2}{2}}\right)\right) = \int q(z)\left(\log \frac{1}{\sigma}\right) dz + \int \frac{z^2}{2} q(z) dz - \int \frac{(z-\mu)^2}{2\sigma^2} q(z)$ 观察第一项就是常数和概率密度积分求和 观察最后一项，其实就是求方差，因此可以很快得到答案 $\frac{1}{2}$
$= \left(\log \frac{1}{\sigma}\right) + \int \frac{1}{2}(z-\mu+\mu)^2 q(z) dz - \frac{1}{2}$
$= \left(\log \frac{1}{\sigma}\right) + \frac{1}{2}\left(\int (z-\mu)^2 q(z) dz + \int \mu^2 q(z) dz + 2\int (z-\mu)(\mu) dz\right) - \frac{1}{2}$ 观察最后一项积分项，是求期望的公式，因此结果为0 综上可以得到结果 $D_{KL}\left(q_\phi(z \mid x) || p_\theta(z)\right) = \left(\log \frac{1}{\sigma}\right) + \frac{\sigma^2+\mu^2}{2} - \frac{1}{2}$ 另一项
$E_z\left[\log(p_\theta(x \mid z))\right]$, 是关于x的后验概率的对数似然，在VAE 中并不对decoder做太强的假设，一般通过一个神经网络来得到正态分 布的均值和方差，因此这一项不能通过解析求出，所以采用采样的方式：
$E_z\left[\log(p_\theta(x \mid z))\right] = \frac{1}{L} \sum_{j=1}^{L} \log p_\theta\left(x^i \mid z^j\right)$

最后，我们从auto-encoder的角度来理解VAE，下图给出了VAE训练的时候的网络结构（以实值样本为例，**注意下面两个图中的 ϵ 节点并不是bias！而是噪声变量，它的维数与 z 相同。**）：

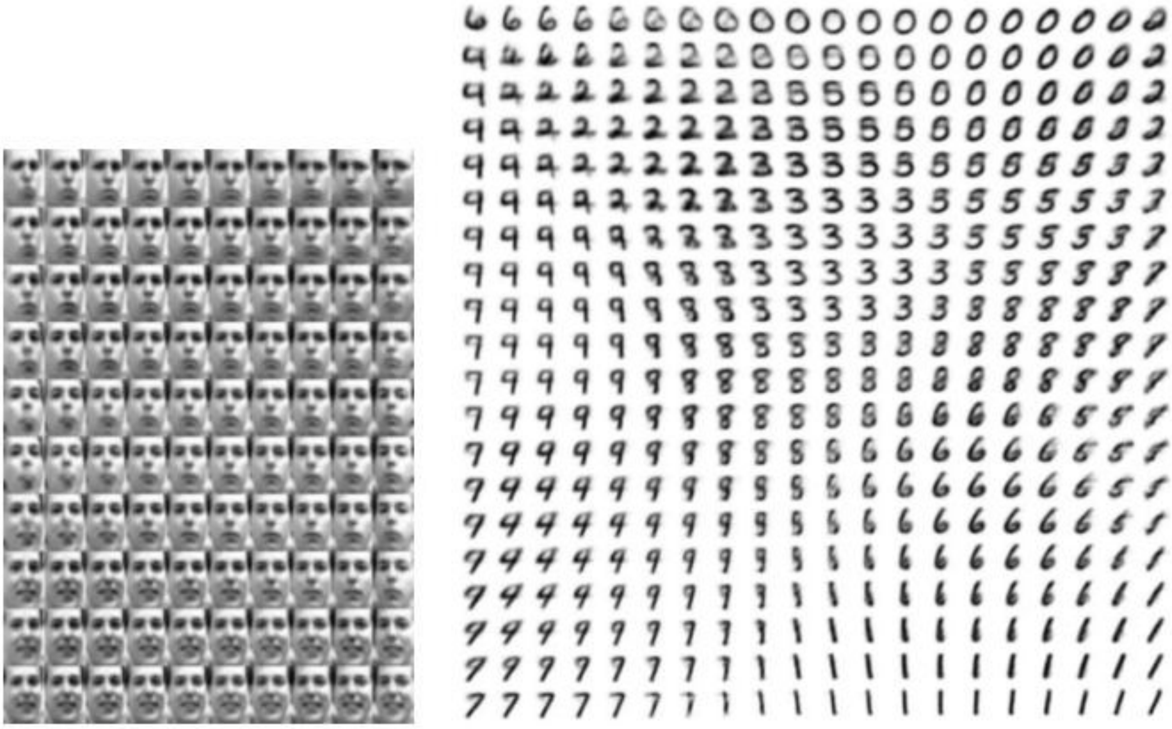


训练好了以后，生成样本采用下面的网络结构：



VAE实验效果

作者在Frey face数据集和MNIST数据集上进行实验，实验得到的数据流形分布如下图所示，可以看出，VAE能够捕捉到图像的结构变化（倾斜角度、圈的位置、形状变化、表情变化等）。这也是VAE的一个好处，它有显式的分布，能够容易地可视化图像的分布。GANs虽然不具有显式的图像分布，但是可以通过对隐变量的插值变化来可视化图像的分布（参见 DCGAN ）。



(a) Learned Frey Face manifold　　(b) Learned MNIST manifold

VAE在不同维数的隐变量空间（z）下生成手写数字的效果如下：

(a) 2-D latent space　(b) 5-D latent space　(c) 10-D latent space　(d) 20-D latent space

可以看出，采用MLP也能产生效果还不错的数字，有趣的是，隐变量维数较低时，生成的图像笔画清晰，但是带有较大的噪声（模糊）；隐变量维数高时，生成的数字部分笔画不清晰，但噪声小。

代码

VAEs网上的代码很多，下面给了三个基于原始论文[1]的代码，作者修改了激活函数和优化方法以取得更好的收敛性。第四个代码是caffe版本，基于文献[2]。

Tensorflow版本： [y0ast/VAE-TensorFlow: Implementation of a Variational Auto- Encoder in TensorFlow]()

Torch版本： [y0ast/VAE-Torch: Implementation of Variational Auto-Encoder in Torch7]()

Theano版本： [y0ast/Variational-Autoencoder: Implementation of a variational Auto-encoder]()

Caffe版本： [Tutorial on Variational Autoencoders]()

参考文献

[1]. Kingma D P, Welling M. Auto-Encoding Variational Bayes[J]. stat, 2014, 1050: 10.

[2]. DOERSCH C. Tutorial on Variational Autoencoders[J]. stat, 2016, 1050: 13.

[3]. Blei, David M., "Variational Inference." Lecture from Princeton,
https://www. cs.princeton.edu/course s/archive/fall11/cos597C/lectures/variational-inference-i.pdf .