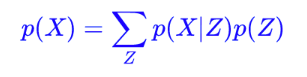
VAE:

本质：进行分布之间的变化，*X*=*g*(*Z*)将原来的概率分布映射到训练集的概率分布，假设Z服从标准正太分布,我们可以改换一下P(X)的写法：



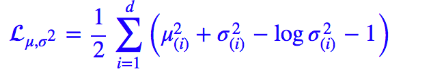
P(x)是我们要求得分布，P(Z)是标准正太分布，P(X|Z)是条件正太分布

其实，**在整个 VAE 模型中，我们并没有去使用** *p***(***Z***)（先验分布）是正态分布的假设，我们用的是假设** *p***(***Z***|***X***)（后验分布）是正态分布**

因为我们后面要训练一个生成器 *X*=*g*(*Z*)，希望能够把从分布 *p*(*Z*|*Xk*) 采样出来的一个 *Zk* 还原为 *Xk*。实现重构得样本与原样本一一对应，这样计算得loss才会由意义。

这样，每一个X样本对应一个正态分布，关于*p*(*Z*|*Xk*)得方差和均值，我们利用神经网络来拟合，形如y=wx+b的形式，把XK转换成符合均值为u,方差为*σ*^2的正太分布，其中，均值和方差通过两个不同的神经网络来拟合。

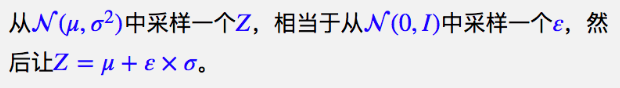
我们通过在XK的分布函数里进行采样然后通过生成器*Xk*=*g*(*Zk*)来得到重构的X，但这个过程存在噪声影响（其实就是正态分布的方差），方差越小，重构loss越小，但如果方差减小至零，无论怎么采样都只会是同一个数，重构出来的X是唯一的，因此丧失了我们的根本目标：让模型有生成能力。因此，我们为了保证让模型既有生成能力，又能重构得好X，我们让分布 *p*(*Z*|*Xk*)向N(0,1)靠拢，这样方差就不会减少至0. 其中我们用KL散度来判定这两个分布的相似程度：



D 为Z的维度（各分量独立的）正态分布与标准正态分布的 KL 散度

我们新加散度的loss，来优化正太分布函数的均值和方差。

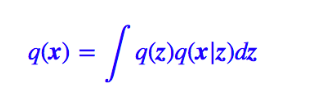
我们要优化重构loss ，就得去不断改变encoder和decoder神经网络的参数，decoder中的参数自然是可以直接利用SGD算法来优化（因为重构的X是由decoder直接算出来得），但由于我们在对Z进行操作的时候，是直接通过采样来获取Z的，与*p*(*Z*|*Xk*)中的u和*σ*^2似乎毫无关系，这里我们通过重参数技巧来解决这个问题：



这样，就建立了与encoder中网络参数的联系，可以优化参数了。

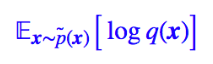
Flow:

对于X，我们只能写出各分量独立的高斯分布，这显然只是众多连续分布中极小的一部分，显然是不够用的。为了解决这个困境，我们通过积分来创造更多的分布：

（1）

q(z) 一般是标准的高斯分布，而 qθ(x|z) 可以选择任意的条件高斯分布或者狄拉克分布。这样的积分形式可以形成很多复杂的分布。理论上来讲，它能拟合任意分布。

我们利用最大似然函数L来求解参数θ：

（2）

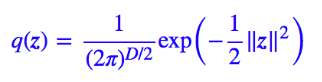
这里，VAE 避开了求解带积分形式的这个困难。VAE 没有直接优化目标 (2)，而是优化一个更强的上界，这使得它只能是一个近似模型，无法达到良好的生成效果.

而flow做出的改变，就是直接把积分算出来！！！！

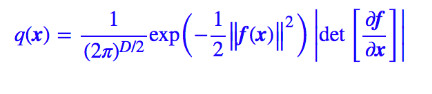
flow 模型选择 q(x|z) 为狄拉克分布 δ(x−g(z))，而且 g(z) 必须是可逆



那么通过 (1) 算 q(x) 相当于是对 q(z) 做一个积分变换 z=f(x)：



令：z=f(x)，且多出了一个**“雅可比行列式”的绝对值：**

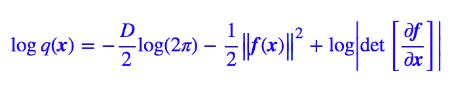


要去对于函数f:

1. 可逆(积分能算)，并且易于求逆函数（它的逆 g 就是我们希望的生成模型）；

2. 对应的雅可比行列式容易计算(减少运算量)。

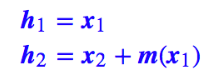
似然函数（优化目标）：



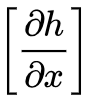
这个优化目标是可以求解的。并且由于 f 容易求逆，因此一旦训练完成，我们就可以随机采样一个 z，然后通过 f 的逆来生成一个样本https://mmbiz.qpic.cn/mmbiz_png/VBcD02jFhglmO52YDDeJfeFk6ic0mwvU3xjRg69SJszqCO26cL5Op9vVv9iaMI0K0iaBgajjdfkTiaP3sqqPBdiaRYw/640?wxfrom=5&wx_lazy=1&wx_co=1，这就得到了生成模型。

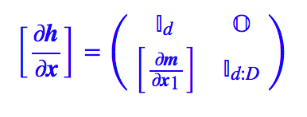
NICE:

对于前面分行列式容易计算，NICE提出了加性耦合层：

（7）

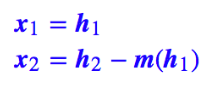
x1,x2 是 x 的某种划分（一般为奇偶行划分），m 是 x1 的任意函数。也就是说，将 x 分为两部分，然后按照上述公式进行变换，得到新的变量 h

变换的雅可比矩阵是一个三角阵，而且对角线全部为 1，用分块矩阵表示为：



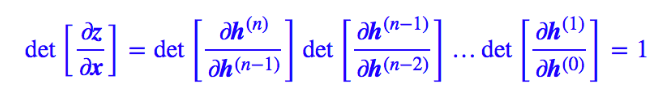
这样一来，这个变换的雅可比行列式为 1，其对数为 0，这样就解决了行列式的计算问题。

同时，(7) 式的变换也是可逆的，其逆变换为：

这样，解决了可逆和行列式难计算的问题

这里,函数m(x)，就把X2部分从线性变成了非线性，但X1仍然是线性的，我们可以通过叠加这种加性耦合层来增强整个X的强非线性：





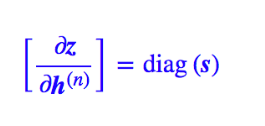
注意：每次变换之后，我们都要反转h1,h2部分，以达到充分混合信息的目的，同时，也让h1部分不再是恒等变换。

为解决维度浪费问题，我们引入了尺度变换层，对最后编码出来的每个维度的特征都做了个尺度变换：

https://mmbiz.qpic.cn/mmbiz_png/VBcD02jFhglmO52YDDeJfeFk6ic0mwvU3SjVI2f3GEFoUZ9Q7eMLZKs8qWbhZZRr0xjHD4WT83c50cELM0f2f9w/640?wxfrom=5&wx_lazy=1&wx_co=1

其中 s=(s1,s2,…,sD) 也是一个要优化的参数向量（各个元素非负）。这个 s 向量能识别该维度的重要程度（越小越重要，越大说明这个维度越不重要，接近可以忽略。），起到压缩流形的作用

尺度变换层的雅可比行列式就不再是 1 了，可以算得它的雅可比矩阵为对角阵：

我们开始设 z 的先验分布为标准正态分布，也就是各个方差都为 1，我们可以将先验分布的方差也作为训练参数，这样训练完成后方差有大有小，方差越小，说明该特征的“弥散”越小，如果方差为 0，那么该特征就恒为均值 0，该维度的分布坍缩为一个点，于是这意味着流形减少了一维。（ 通过变换得到si=1/σi）;

尺度变换：衡量一个特征的重要程度，在nice中，尺度值使标准差的倒数，方差越小，该值越大，该特征的值越不重要。

先验分布选为各分量独立的高斯分布的好处：

**一个好的特征，理想情况下各个维度之间应该是相互独立的，这样实现了特征的解耦，使得每个维度都有自己独立的含义，**反过来，由于 z 的每个维度的独立性，理论上我们控制改变单个维度时，就可以看出生成图像是如何随着该维度的改变而改变，从而发现该维度的含义

这样，我们就能理解“先验分布为各分量独立的高斯分布”的好处了，由于各分量的独立性，我们有理由说当我们用f对原始特征进行编码时，输出的编码特征 z=f(x) 的各个维度是解耦的。

REAL NVP:

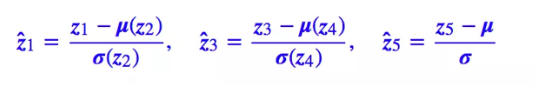
NICE的神经网络只是使用了简单的全连接层去训练，没有用到卷积的使用

1. 在耦合层中用卷积代替了全连接层

这使得我们可以更好地处理图像问题，并且减少参数量，还可以更充分发挥模型的并行性能

1. 提出多尺度层，降低运算量，提供强大的正则想过

既减少了模型复杂度、又提升了结果的策略。原始输入经过第一步 flow 运算（“flow 运算”指的是多个仿射耦合层的复合）后，输出跟输入的大小一样，这时候将输入对半分开两半 z1,z2（自然也是沿着通道轴），其中 z1 直接输出，而只将 z2 送入到下一步 flow 运算，作为不同位置的多尺度输出，z1,z3,z5 的地位是不对等，改变一下：



然后认为 [ẑ1,ẑ3,ẑ5] 服从标准正态分布。同 NICE 的尺度变换层一样，这三个变换都会导致一个非 1 的雅可比行列式，也就是要往 loss 中加入形如https://image.jiqizhixin.com/uploads/editor/5183dfc0-2916-4cbd-9da7-7bde544aae5c/1535347013302.png的这一项。

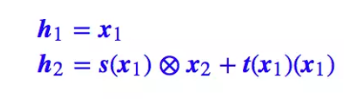
乍看之下多尺度结构就是为了降低运算量，但并不是那么简单。由于 flow 模型的可逆性，输入输出维度一样，事实上这会存在非常严重的维度浪费问题，这往往要求我们需要用足够复杂的网络去缓解这个维度浪费。

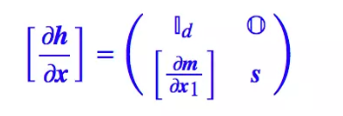
**多尺度结构相当于抛弃了 p(z) 是标准正态分布的直接假设，而采用了一个组合式的条件分布**，这样尽管输入输出的总维度依然一样，但是不同层次的输出地位已经不对等了，模型可以通过控制每个条件分布的方差来抑制维度浪费问题（极端情况下，方差为 0，那么高斯分布坍缩为狄拉克分布，维度就降低 1），条件分布相比于独立分布具有更大的灵活性。而如果单纯从 loss 的角度看，多尺度结构为模型提供了一个强有力的正则项（相当于多层图像分类模型中的多条直连边）。

1. X1,x2的简单反转改为固定的随机排序

更充分混合信息，最终的 loss 可以更低，由于我们是对图像操作，为保留空间局部相关性，RealNVP 约定分割和打乱操作，都只对“通道”轴执行。

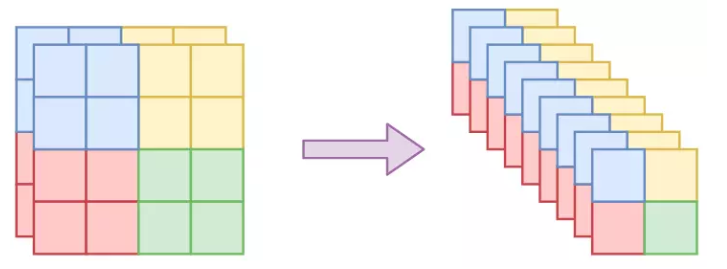
1. 由仿射耦合层代替NICE中加性耦合层。





仿射耦合层使得行列式的值不再为1，行列式的意义可以理解为体积的大小RealNVP = real-valued non-volume preserving 实值非体积保持。

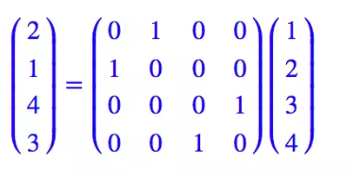
1. 引入squeeze操作



有了 squeeze 这个操作，我们就可以增加通道轴的维数，但依然保留局部相关性，从而我们前面说的所有事情都可以进行下去了，所以 squeeze 成为 flow 模型在图像应用中的必备操作

Glow:

1. 在全连接中，使用置换矩阵打乱X顺序替代REAL NVP中的随机打乱，在卷积中，使用1\*1矩阵在通道方向上打乱顺序。





但这样权重矩阵W的行列式变得很难计算，不是对角矩阵

方法：利用LU分解，L 是一个下三角阵，对角线元素全为 1；U 是一个上三角阵，P 是一个置换矩阵 W为正交矩阵

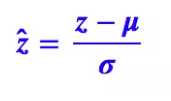


这样，行列式变得可以计算：



这就是 Glow 中给出的技巧：先随机生成一个正交矩阵，然后做 LU 分解，得到 P,L,U，固定 P，也固定 U 的对角线的正负号，然后约束 L 为对角线全 1 的下三角阵，U 为上三角阵，优化训练 L,U 的其余参数。

2，提出Actnorm来缩放平移输入参数X



也就是标准化随机变量，其中 μ,σ 都是训练参数。**由于 Actnorm 的存在，仿射耦合层的尺度变换已经显得不那么重要了**。相比于加性耦合层，仿射耦合层多了一个尺度变换层，从而计算量翻了一倍。但事实上相比加性耦合，仿射耦合效果的提升并不高（尤其是加入了 Actnorm 后），**所以要训练大型的模型，为了节省资源，一般都只用加性耦合**