变分自动编码器教程

卡尔·多尔施
卡内基梅隆/加州大学伯克利分校
2016年8月16日

摘要

在短短三年内，变分自动编码器（VAE）已经成为无人监督学习复杂分布的最流行的方法之一。VAE很有吸引力，因为它们建立在标准函数逼近器（神经网络）之上，并且可以用随机梯度下降进行训练。VAE已经在生成多种复杂数据方面表现出了希望，包括手写数字[1, 2]，面[1, 3, 4]，门牌号码[5, 6]，CIFAR图像[6]，场景的物理模型[4]，分割[7]，并从静态图像预测未来[8]. 本教程介绍了VAE背后的直觉，解释了它们背后的数学，并描述了一些经验行为。没有假设变分贝叶斯方法的先验知识。

*关*键词：变分自动编码器，无监督学习，结构化预测，神经网络

1 介绍

“生成建模”是机器学习的一个广泛领域，它处理分布模型P（X），在一些潜在的高维空间中在数据点X上定义。例如，图像是一种流行的数据，我们可以为其创建生成模型。X每个“数据点”（图像）具有数千或数百万个维度（像素），并且生成模型的工作是以某种方式捕获像素之间的依赖性，例如，附近的像素具有相似的颜色，并且被组织成对象。“捕获”这些依赖关系的确切含义取决于我们想要对模型做什么。一种直接的生成模型简单地允许我们在数值上计算P（X）。在图像的情况下，X值

看起来像真实图像应该获得高概率，而看起来像随机噪声的图像应该得到低概率。然而，像这样的模型并不一定有用：知道一个图像不太可能无法帮助我们合成一个可能的图像。

相反，人们经常关心的是生成更多的示例，这些示例与数据库中已有的示例相似，但并不完全相同。我们可以从原始图像数据库开始，并合成新的，看不见的图像。我们可能会接收一个像植物这样的3D模型的数据库，并生成更多的模型来填充视频游戏中的森林。我们可以采取手写文本，并尝试生成更多的手写文本。像这样的工具实际上可能对图形设计师有用。我们可以通过说我们得到根据一些未知分布$P_{燃气轮机}$（X）分布的示例X来形式化这个设置，并且我们的目标是学习我们可以从中采样的模型P，使得P尽可能相似。 $P_{燃气轮机}$。

训练这种类型的模型一直是机器学习社区中长期存在的问题，而且经典地说，大多数方法都有三个严重的缺点之一。首先，他们可能需要对数据结构进行强有力的假设。其次，他们可能会进行严格的近似，导致模型不理想。

或者第三，他们可能依赖于计算成本昂贵的推理程序，如Markov Chain Monte Carlo。最近，一些工作在通过反向传播训练神经网络作为强大的函数逼近器方面取得了巨大进展[9].这些进步产生了有希望的框架，可以使用基于反向传播的函数逼近器来构建生成模型。最受欢迎的此类框架之一是变分自动编码器[1,3]，本教程的主题。这种模型的假设很弱，通过反向传播训练很快。VAE确实做了近似，但是这种近似引入的误差在高容量模型下可以说是很小的。这些特征促成了这一点他们受欢迎程度迅速提升。

本教程旨在成为对VAE的非正式介绍，而不是关于它们的正式科学论文。它针对的是那些可能用于生成模型的人，但可能不具备变异贝叶斯方法和VAE所基于的“最小描述长度”编码模型的强大背景。本教程开始于加州大学伯克利分校和卡内基梅隆大学的计算机视觉阅读小组的演示，因此偏向于视觉观众。感谢您的改进建议。

1.1 预备：潜变量模型

在训练生成模型时，维度之间的依赖关系越复杂，模型训练就越困难。例如，生成手写字符图像的问题。简单地说，我们只关心数字0-9的建模。如果角色的左半部分包含5的左半部分，则右半部分不能包含0的左半部分，或者角色非常清楚地看起来不像任何真实的数字。直观地说，如果模型在为任何特定像素指定值之前首先确定要生成哪个字符，它会有所帮助。这种决定被正式称为潜在变量。也就是说，在我们的模型绘制任何东西之前，它首先从集合[0，...，9]中随机采样数字值z，然后确保所有笔划都匹配该字符。z被称为'潜在'，因为只给出模型产生的字符，我们不一定知道潜在变量的哪些设置产生了字符。我们需要使用类似计算机视觉的东西来推断它。

在我们可以说我们的模型代表我们的数据集之前，我们需要确保对于数据集中的每个数据点X，潜在变量有一个（或许多）设置，这会导致模型生成与X非常相似的东西。正式地说，我们在高维空间中有一个潜在变量z的向量，我们可以根据定义的概率密度函数（PDF）P（z）轻松地进行采样。然后，假设我们有一系列确定性函数f（z;θ），参数化通过某个空间Θ中的向量θ，其中f：$Z \times \Theta \to X$.f是确定性的，但如果z是随机的并且θ是固定的，则f（z;θ）是空间中的随机变量X.我们希望优化θ，使得我们可以从P（z）中采样z，并且在高概率下，f（z;θ）将类似于我们数据集中的X.

为了使这个概念在数学上精确，我们的目标是在整个生成过程中最大化训练集中每个X的概率，根据：

$$P(X) = \int P(X \mid z;\theta)\, P(z)\, dz。 \qquad (1)$$

这里，f（z;θ）已被分布P（X z;θ）所取代，这使得我们可以通过使用总概率定律使X对z的依赖性明确。这个框架背后的直觉 - 称为"最大似然" - 是如果模型可能产生训练集样本，那么它也可能产生类似的样本，并且不太可能产生不同的样本。在VAE中，该输出分布的选择通常是高斯分布，即P（X z;θ）= $N(X; f(z;\theta), \sigma^2 I)$。也就是说，它具有均值f（z;θ）和协方差等于单位矩阵I乘以某个标量σ（它是超参数）。这种替换对于形成某些z的直觉是必要的

*

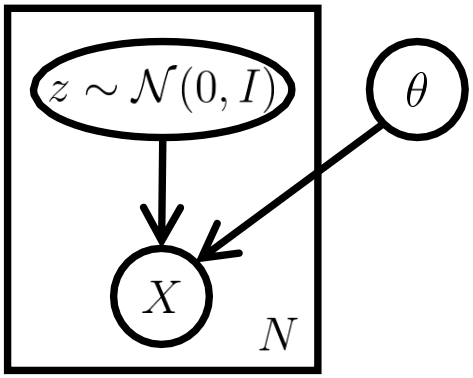


图1：标准VAE模型表示为图形模型。注意任何结构或甚至“编码器”路径的显着缺失：可以在没有任何输入的情况下从模型中进行采样。这里，矩形是“板表示法”，这意味着我们可以在z和XN时间采样，而模型参数θ保持固定。

需要产生仅仅像X一样的样本。一般来说，特别是在训练的早期，我们的模型不会产生与任何特定X相同的输出。通过具有高斯分布，我们可以使用梯度下降（或任何其他优化）技术）通过使f（z;θ）接近某些z的X来增加P（X），即在生成模型下逐渐使训练数据更可能。如果P（X z）是狄拉克δ函数，这将是不可能的，因为如果我们确定性地使用X = f（z;θ）将会是这样！注意，输出分布不需要是高斯分布：例如，如果X是二进制的，那么P（X z）可能是由f（z;θ）参数化的伯努利。重要的特性是简单地可以计算P（X z），并且在θ中是连续的。从这里开始，我们将从f（z;θ）中省略θ以避免混乱。

2 变分自动编码器

VAE的数学基础实际上与经典自动编码器相关性很小，例如稀疏自动编码器[10, 11]或去噪自动编码器[12, 13].VAE近似最大化方程1，根据图中所示的模型1. 它们被称为“自动编码器”，因为源自此设置的最终训练目标确实具有编码器和解码器，并且类似于传统的自动编码器。与稀疏自动编码器不同，通常没有类似于稀疏性惩罚的调整参数。与稀疏和去噪自动编码器不同，我们

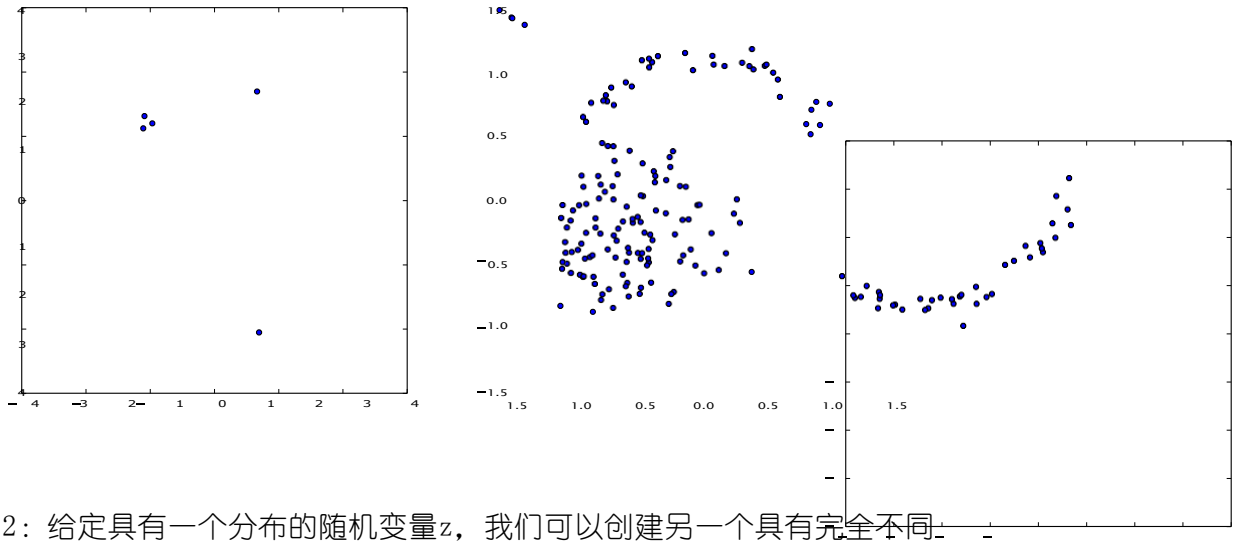
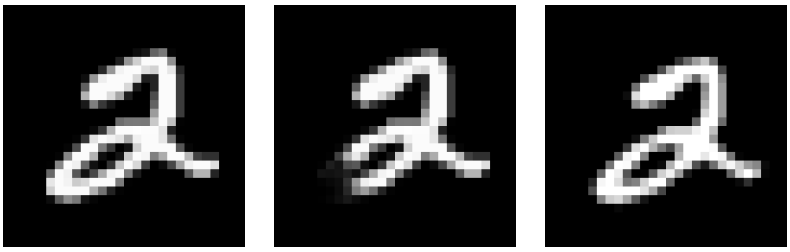


图2：给定具有一个分布的随机变量z，我们可以创建另一个具有完全不同分布的随机变量X = g（z）。左：来自高斯分布的样本。右：通过函数g（z）= z / 10 + z / z映射的那些相同样本形成环。这是VAE用于创建任意分布的策略：确定性函数g是从数据中学习的。

可以直接从P（X）采样（不执行马尔可夫链蒙特卡罗，如[14]).

解决方程1，VAE必须处理两个问题：如何定义潜在变量z（即，决定它们代表什么信息），以及如何处理z上的积分。VAEs给出了两者的明确答案。

首先，我们如何选择潜在变量z，以便捕获潜在信息？回到我们的数字示例，模型在开始绘制数字之前需要做出的“潜在”决定实际上相当复杂。它不仅需要选择数字，还需要选择绘制数字的角度，笔划宽度以及抽象风格属性。更糟糕的是，这些属性可能是相关的：如果一个写入速度更快，则可能会产生更多角度的数字，这也可能导致更细的笔划。理想情况下，我们希望避免手动决定z编码的每个维度的哪些信息（尽管我们可能希望手动指定某些维度[4]).我们还想避免明确地描述依赖关系 - 即z的维度之间的潜在结构。VAE采取了一种不寻常的方法来解决这个问题：他们认为没有简单的解释
对于z的维度，并且断言z的样本可以从简单分布中绘制，即N（0，I），其中I是单位矩阵。怎么样



(a) (b) (c)

图3：仅使用采样很难测量模型下图像的可能性。给定图像X（a），中间样本（b）在欧几里德距离上比右侧（c）更接近。因为像素距离与感知距离是如此不同，所以在可以认为X可能在模型下的证据之前，样本需要在与数据点X的像素距离上非常接近。

这可能吗？关键是要注意d维中的任何分布都可以通过采用一组正常分布的d变量并通过一个足够复杂的函数映射它们来生成。[1]. 例如，假设我们想构建一个2D随机变量，其值位于环上。如果z是2D并且是正态分布的，则g（z）= z / 10 + z / z大致为环形，如图2. 因此，提供了强大的函数逼近器，我们可以简单地学习一个函数，它将我们独立的，正态分布的值映射到模型可能需要的任何潜在变量，然后将这些潜在变量映射到X. 事实上，回想一下P（ X z;θ）=（X f（z;θ），σ^2 I）。如果f（z; θ ）是一个多层神经网络，那么我们可以想象网络使用它的前几层将正态分布的z映射到潜在值（如数字标识，笔画粗细，角度等）完全正确的统计数据。然后，它可以使用后面的图层将这些潜在值映射到完全呈现的数字。通常，我们不需要担心确保潜在结构存在。如果这种潜在结构有助于模型准确地再现（即最大化）训练集的可能性，那么网络将在某一层学习该结构。

现在剩下的就是最大化方程式1，其中P（z）= $\mathcal{N}(z|0,I)$. 在机器学习中很常见，如果我们能找到P（X）的可计算公式，并且我们可以采用该公式的梯度，那么我们可以优化

[1]在一个维度中，您可以使用由高斯CDF组成的所需分布的逆累积分布函数（CDF）。这是“逆变换采样”的扩展。对于多维度，所述过程从单个维度的边际分布开始，并重复每个附加维度的条件分布。参见Devroye等人的“反演方法”和“条件分布方法”。[15]

使用随机梯度上升来模拟模型。实际上在概念上直接计算P（X）：我们首先采样大量z值z_1，...，z_n 和计算P（X） $\frac{1}{n}\sum_i P$（X z_i）。这里的问题是在高维空间中，在我们准确估计P（X）之前，n可能需要非常大。要了解原因，请考虑我们的手写数字示例。假设我们的数字数据点存储在像素空间中，在28x28图像中，如图所示3. 由于P（X z）是各向同性高斯，因此X的负对数概率是f（z）和X之间的欧几里德距离的比例平方。如图所示3(a) 是目标
（X）我们试图找到P（X）。一个产生的模型
图片如图所示3(b) 可能是一个糟糕的模型，因为这个数字不太像2. 因此，我们应该设置我们的高斯分布的σ超参数，使得这种错误的数字对P（X）没有贡献。另一方面，产生图的模型3(c) （与X相同但向下移动并向右移动半个像素）可能是一个很好的模型。我们希望这个样本能为P（X）做出贡献。不幸的是，我们不能两种方式：X和图之间的平方距离3(c) 是.2693（假设像素范围在0和1之间），但在X和图之间3(b) 它只是.0387。这里的教训是为了拒绝像图这样的样本3(b)，我们需要将σ设置得非常小，这样模型需要生成比X更像X的东西3(c)! 即使我们的模型是一个精确的数字生成器，我们可能需要在生成一个与图中的数字足够相似的2之前对数千个数字进行采样3(a). 我们可以通过使用更好的相似性度量来解决这个问题，但实际上这些很难在视觉等复杂领域中进行设计，并且如果没有标签指示哪些数据点彼此相似，则很难进行训练。相反，VAE改变采样程序以使其更快，而不改变相似性度量。

2.1 设定目标

在使用抽样来计算方程式时，我们可以采用一种捷径1? 在实践中，对于大多数z，P（X z）将几乎为零，因此对我们对P（X）的估计几乎没有贡献。变分自动编码器背后的关键思想是尝试对可能产生X的z进行采样，并从中计算P（X）。这意味着我们需要一个新的函数Q（z X），它可以取X的值并给出一个可能产生X的z值的分布。希望可能在Q下的z值的空间将小得多而不是所有z的空间
可能在先前的P（z）之下。这让我们，例如，计算$E_{z\sim Q}$P（X z）
相对容易。但是，如果从任意分布中采样z

使用PDF Q（z），它不是N（0，I），那么它如何帮助我们优化P（X）？我们需要做的第一件事是关联$E_{z\sim Q}$P（X z）和P（X）。我们稍后会看到Q的来源。

$E_{z\sim Q}$P（X z）和P（X）之间的关系是变分贝叶斯方法的基石之一。我们从P（z X）和Q（z）之间的Kullback-Leibler散度（KL散度或）的定义开始，对于某些任意Q（可能或可能不依赖于X）：

$$D[Q(z)\,P(z\mid X)] = E_{z\sim Q}[\log Q(z) - \log P(z\mid X)]。 \quad (2)$$

通过将贝叶斯规则应用于P（z｜X），我们可以将P（X）和P（X｜z）都得到这个等式：

$$D[Q(z)\,P(z\mid X)] = E_{z\sim Q}[\log Q(z) - \log P(X\mid z) - \log P(z)] + \log P(X)。 \quad (3)$$

log P（X）出于期望，因为它不依赖于z.否定双方，重新安排和将$E_{z\sim Q}$的一部分收缩为KL-分歧条款产生：

$$\log P(X) - D[Q(z)\,P(z\mid X)] = E_{z\sim Q}[\log P(X\mid z)] - D[Q(z)\,P(z)]。 \quad (4)$$

注意X是固定的，Q可以是任何分布，而不仅仅是一个能够很好地将X映射到可以产生X的z的分布。由于我们有兴趣推断P（X），所以构造一个是有意义的。 Q哪个

取决于X，特别是使D［Q（z）｜P（z｜X）］小的一个：

$$\log P(X) - D[Q(z\mid X)\,P(z\mid X)] = E_{z\sim Q}[\log P(X\mid z)] - D[Q(z\mid X)\,P(z)]。 \quad (5)$$

这个等式服务是变分自动编码器的核心，值得花一些时间思考它所说的内容[2].在两个句子中，左侧有我们想要最大化的数量：log P（X）（加上一个误差项，这使得Q产生可以再现给定X的z;如果Q是高容量，这个项将变小）。右手边是我们可以通过随机梯度下降优化的东西，给出正确的Q选择（虽然它可能不是很明显但是如何）。注意框架 - 特别是方程式的右侧5，具有突然采取了一个看起来像自动编码器的形式，因为Q将X“编码”为z，而P正在“解码”它以重建X.我们稍后将更详细地探讨这种连接。

[2]历史上，这个数学（特别是方程式）5）早在VAE之前就知道了。例如，亥姆霍兹机器[16]（见公式5）使用几乎相同的数学，但有一个关键的区别。我们期望中的积分被Dayan等人的总和取代。[16]，因为亥姆霍兹机器假设潜在变量的离散分布。此选择可防止在VAE中使梯度下降易于处理的转换。

现在了解有关Equatinon的更多细节5. 从左侧开始，
我们最大化log P (X)，同时最小化 $D[Q(z|X)|P(z|X)]$，P (z X) 不是我们可以分析计算的东西：它描述了val-
在我们的模型中，z可能会产生像X这样的样本1. 然而，左边的第二项是拉Q (zx) 以匹配P (z X)。假设我们对Q (zx) 使用任意高容量模型，那么Q (zx) 有望实际匹配P (z X)，在这种情况下，这个KL-发散项将为零，我们将直接优化log P (X)。作为一个额外的好处，我们已经使难以处理的P (z | X)易于处理：我们可以
只需使用Q (z | x) 来计算它。

2.2 优化目标

那么我们如何在方程的右边进行随机梯度下降5? 首先，我们需要更加具体地了解Q (z X) 将采用的形式。通常的选择是Q (z X) = (zμ (X;θ)，Σ (X;θ))，其中μ和Σ是任意确定性函数，参数θ可以从数据中学习 (我们将在后面的等式中省略θ)。在实践中，μ和Σ再次通过神经网络实现，并且Σ被约束为对角矩阵。这种选择的优点是计算，因为它们清楚地说明了如何计算右手侧。最后一项 $-[Q(z|X)P(z)]$ - 现在是两个多元高斯分布之间的KL-发散，可以以封闭形式计算如下：

$$D[N(\mu_0, \Sigma_0)\, N(\mu_1, \Sigma_1)] = \frac{1}{2}\left[\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^t \Sigma^{-I} (\mu_1 - \mu_0) - k + \log\right] \quad (6)$$

(6)

其中k是分布的维数。在我们的例子中，这简化为：

$$D[N(\mu(X), \Sigma(X))\, N(0, I)] = \frac{1}{2}\left[\operatorname{tr}(\Sigma(X)) + (\mu(X))^t (\mu(X)) - \cdot\right] \quad (7)$$

方程右边的第一项5 有点棘手。我们可以使用抽样来估计$E_{z\sim Q}[\log P(X|z)]$，但是得到一个好的估计需要传递许多z到f的样本，
很贵。因此，作为随机梯度下降的标准，我们取z的一个样本并将该z的P (X | z) 视为近似值
$E_{z\sim Q}[\log P(X|z)]$。毕竟，我们已经在做随机梯度下降
从数据集D中采样的不同X值。我们的完整方程

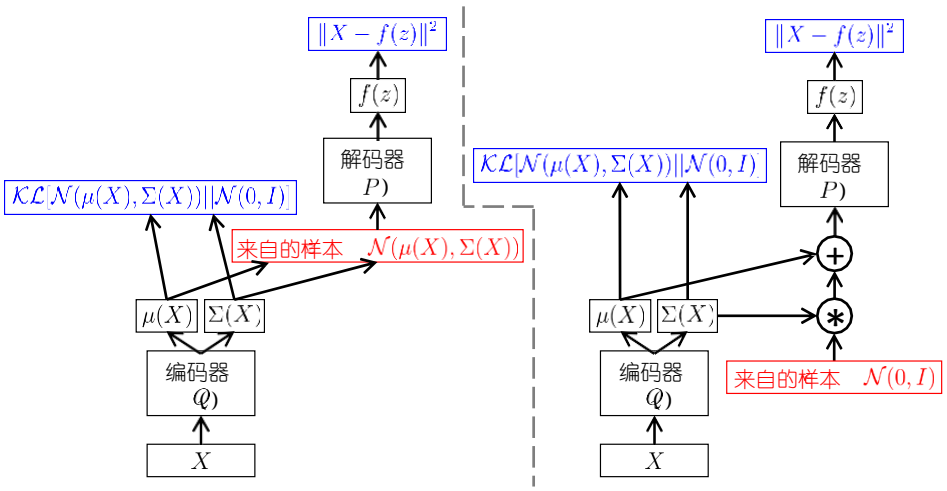


图4：作为前馈神经网络实现的训练时变分自动编码器，其中P（X z）是高斯分布。左边没有“重新参数化技巧”，正确就是它。红色显示不可微分的采样操作。蓝色显示损失层。这些网络的前馈行为是相同的，但反向传播只能应用于正确的网络。

想要优化的是：

$$E_{x\sim d}[\log P(X) - D[Q(z|X)P(z|X)]] = E_{x\sim d}[E_{z\sim Q}[\log P(X|z)] - D[Q(z|X)P(z)]]。 \quad (8)$$

如果我们采用该等式的梯度，则可以将梯度符号移动到期望值中。因此，我们可以从分布Q（z | X）中采样单个X值和单个z值，并计算以下梯度：

$$\log P(X|z) - D[Q(z|X)P(z)]。 \quad (9)$$

然后我们可以在任意多个X和z样本上平均该函数的梯度，并且结果收敛到方程的梯度8.

然而，方程式存在严重问题9. $E_{z\sim Q}[日志\ P(X\ z)]$ 不仅取决于P的参数，还取决于Q的参数。

但是，在方程式中9,这种依赖已经消失了！为了使VAE工作，必须驱动Q来生成P可以可靠解码的X代码。要以不同的方式查看问题，网络中描述的方程式9 很像图中所示的网络4 （剩下）。该网络的正向传递工作正常，如果输出在X和z的许多样本上取平均值，则产生正确的期望值。但是，我们需要

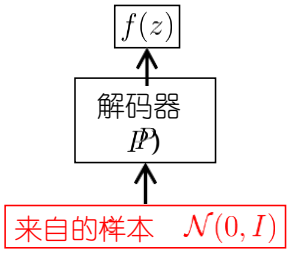


图5：测试时变分“自动编码器”，它允许我们生成新的样本。简单地丢弃“编码器”路径。

通过从Q（z X）采样z的层反向传播误差，这是一个非连续的操作并且没有梯度。通过反向传播的随机梯度下降可以处理随机输入，但不能处理网络中的随机单位！解决方案，称为[重新参数化技巧]在[1]，是将采样移动到输入层。给定μ（X）和Σ（X） - Q（z X）的均值和协方差 - 我们可以通过首先采样€（0，I）从（μ（X），Σ（X））进行采样，然后计算z =μ（X）+$\Sigma^{1/2}$（X）€。因此，我们实际采用的梯度是：

$$E_{x\sim d}\, E_{\epsilon\sim n\ (0,i)}[\log P(X \mid z=\mu(X)+\Sigma^{1/2}(X)*)) - D[Q^{*}(z \mid X)P(z)]。 \quad (10)$$

这在图中示意性地示出4（对）。请注意，对于依赖于我们的模型参数的分布，没有任何期望，因此我们可以安全地将渐变符号移动到它们中，同时保持平等。也就是说，给定一个固定的X和€，这个函数在P和Q的参数中是确定性和连续的，这意味着反向传播可以计算出适用于随机梯度下降的梯度。值得指出的是，“重新参数化技巧”仅在我们能够通过评估函数h（η，X）从Q（z X）进行采样时才有效，其中η是来自未学习的分布的噪声。此外，h必须在X中连续，以便我们可以通过它进行反向提升。这意味着Q（z X）（因此P（z））不能是离散分布！如果Q是离散的，那么对于固定的η，要么h需要忽略X，要么需要某个点h（η，X）从Q的样本空间中的一个可能值“跳跃”到另一个，即a间断。

2.3 测试学习的模型

在测试时，当我们想要生成新样本时，我们只需将z（0，I）的值输入到解码器中。也就是说，我们删除了“编码器”，包括可以改变的乘法和加法运算

分布z。这个（非常简单的）测试时网络如图所示5.

假设我们想要评估模型下测试示例的概率。通常，这不容易处理。但请注意，[Q（z X）P（z X）]为正，意味着Equa的右侧重刑5 是P（X）的下限。这个下限仍然不太可能由于期望超过z而以封闭形式计算，这需要采样。然而，从Q中采样z给出了期望的估计，该估计通常比从（0，I）采样z快得多地收敛，如部分所讨论的2. 因此，这个下界可以是一个有用的工具，可以粗略地了解我们的模型捕获特定数据点X的程度。

2.4 解释目标

到目前为止，您有望确信VAE中的学习是易处理的，并且它在整个数据集中优化了log P（X）之类的东西。但是，我们并没有精确优化log P（X），因此本节旨在深入研究目标函数的实际作用。我们讨论三个主题。首先，我们询问除了log P（X）之外，通过优化[Q（z X）P（z X）]引入了多少误差。其次，我们描述了VAE框架 – 尤其是方程的rhs5，在信息理论术语，将其与基于最小描述长度的其他方法联系起来。最后，我们研究VAE是否具有“正则化参数”稀疏自动编码器中的稀疏性惩罚。

2.4.1 D [Q（z | X）P（z | X）]的误差

该模型的易处理性依赖于我们的假设，即Q（z X）可以被建模为具有一些平均μ（X）和方差Σ（X）的高斯。如果仅[Q（z X）P（z X）]变为零，则P（X）收敛（分布）到真实分布。不幸的是，确保发生这种情况并非直截了当。即使我们假设μ（X）和Σ（X）是任意高容量，对于我们用来定义P的任意f函数，后验P（z X）不一定是高斯的。对于固定P，这可能意味着[Q（z X）P（z X）]永远不会变为零。然而，好消息是有无数的f函数导致我们的模型生成任何给定的输出分布。这些函数中的任何一个都将最大化log P（X）。因此，我们所需要的只是一个函数f，它最大化log P（X）并导致P（z X）对于所有X都是高斯的。如果是这样，[Q（z X）P（z X）]将拉动我们的模型朝向分配的参数化。那么，对于我们可能想要近似的所有分布，是否存在这样的函数？我还没有意识到有人证明这一点，但事实证明这一点

可以证明这样的函数确实存在，只要σ相对于地面实况分布的CDF曲率较小（至少在1D中;证据包含在附录中）A). 在实践中，这样小的σ可能导致现有机器学习算法的问题，因为梯度将变得严重缩放。然而，令人欣慰的是，至少在这种情况下，VAE具有零近似误差。这一事实表明，未来的理论工作可能会向我们展示VAE在更实际的设置中有多少近似误差。似乎应该可以在附录中扩展证明技术A 到多个维度，但这留待将来的工作。

2.4.2 信息理论解释

查看方程右侧的另一个重要方法5 就信息理论而言，特别是“最小描述长度”原则，它激发了许多VAE的前辈，如亥姆霍兹机器[16]，唤醒 - 睡眠算法[17]，深信仰网[18]和玻尔兹曼机器[19]. log P（X）可以看作是使用理想编码在我们的模型下构造X所需的总位数。方程式的右侧5 将其视为构造X的两步过程。我们首先使用一些位来构造z。回想一下KL-发散是以比特（或nat）来衡量的。我们使用[Q（z X）P（z）]来测量构造z所需的位，因为在我们的模型中，我们假设从P（z）= （z 0，I）采样的任何z都不包含任何信息关于X. 因此，当z来自Q（z X）而不是来自P（z）时，我们需要测量我们得到的关于X的额外信息量（更多细节，请参阅[后面的“位返回”参数]20, 21]）. 在第二步中，D_{KL}（P（X z）$\|$ N）测量在理想编码下从z重建X所需的信息量。因此，总比特数（log P（X））是这两个步骤的总和，减去我们为Q作为次优编码所支付的代价（[P（z X）Q（z X）]）。请注意，第二步是编码关于X的信息的相当浪费的方式：P（X z）不模拟我们模型下X的维度之间的任何相关性，因此即使理想的编码也必须分别对每个维度进行编码。

2.4.3 VAE和正则化参数

看方程5, 将[Q（z X）P（z）]视为正则化项很有意思，就像稀疏自编码器中的稀疏正则化一样[10]. 从这个角度来看，有趣的是，变量自动编码器是否具有任何“正则化参数”。也就是说，在稀疏自动编码器目标中，我们有一个λ正则化参数。

目标函数看起来像这样：

$$\varphi(\psi(X)) - X^{2} + \lambda\,\psi(X)_{0}$$

(1

1）其中ψ和φ分别是编码器和解码器的功能

″，$_{0}$是L_{0} 范数，它鼓励编码稀疏。该 λ 必须手动设定。

有趣的是，变分自动编码器通常不具有这样的正则化参数，这是好的，因为这是程序员需要调整的一个较少的参数。但是，对于某些模型，我们可以使它看起来像这样的正则化参数存在。这很诱人
认为这个参数可以来自将z~N（0，I）变为类似z^{t}（0，λI）的东西，但事实证明这不会改变模型。至
看到这一点，注意我们可以通过用$f^{t}(z^{t}) = f(z^{t}/\lambda)$，$\mu^{t}(X) = \mu$来表示它们将这个常数吸收到P和Q中（X）$\lambda$，和$\Sigma^{t}(X) = \Sigma(X)\lambda^{2}$。这将产生一个目标函数，其值（方程的右侧）5）与z~N（0，I）的损失相同。另外，模型
因为z^{t}/λ，因此采样X将是相同的 $(0, I)$.

但是，还有另一个可以来自正则化参数的地方。回想P（X z）（f（z），σ^{2} I）：因此，改变该σ将改变P（X z）而不影响[Q（z X）P（z）]。更详细地，log P（X z）= C ^{1}X f（z）$^{2}/\sigma^{2}$，其中C是不依赖于f的常数，因此可以在优化期间被忽略。因此，σ可以被视为等式的rhs的两个项之间的加权因子5. 但请注意，此参数的存在依赖于我们选择给定z的X分布。如果X是二元的并且我们使用伯努利输出模型，那么这个正则化参数就会消失，而将它带回来的唯一方法就是使用像复制X维度那样的黑客攻击。从信息论的角度来看，这是有道理的：当X是二进制，我们实际上可以计算编码X所需的位数，以及方程右侧的两个项5 使用相同的单位来计算这些位。但是，当X是连续的时，每个样本包含无限的信息。我们选择σ决定了我们期望模型重建X的准确程度，这是必要的，以便信息内容可以变得有限。

3 条件变分自动编码器

让我们回到我们生成手写数字的运行示例。假设我们不只是想生成新的数字，而是想要将数字添加到由单个人编写的现有数字串中。这类似于计算机图形学中一个真正实际的问题，称为孔填充：给定

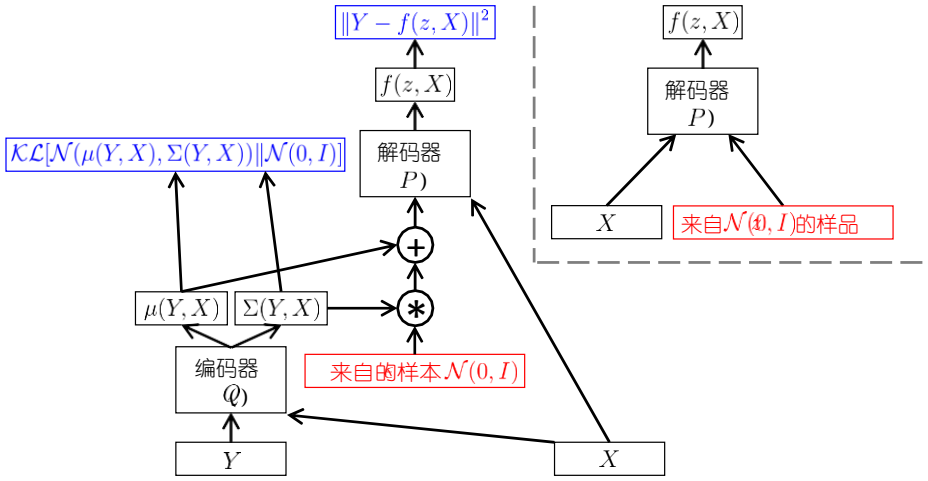


图6：左：训练时条件变分自动编码器实现为前馈神经网络，遵循与图1相同的符号4. 右：当我们想要从中进行采样时，在测试时使用相同的模型 $P(Y|X)$.

用户已移除不需要的对象的现有图像，目标是用看似合理的像素填充孔。两个问题的一个重要困难是合理输出的空间是多模态的：下一个数字或外推像素有很多种可能性。在这种情况下，标准回归模型将失败，因为训练目标通常会惩罚单个预测与地面实况之间的距离。面对这样的问题，回归器可以产生的最佳解决方案是在可能性之间，因为它最小化了预期的距离。在数字的情况下，这很可能看起来像无意义的模糊，它是所有可能数字的“平均图像”以及可能发生的所有可能的样式[3]. 我们需要的是一种算法，它接收一个字符串或一个图像，并产生一个我们可以从中采样的复杂的多模式分布。输入条件变量自动编码器（CVAE）[7, 8]，通过简单地调整输入上的整个生成过程来修改上一节中的数学。CVAE允许我们解决输入到输出映射是一对一的问题

[3]去噪自动编码器[12, 13]可以看作是回归模型的略微概括，可能会改善其行为。也就是说，我们会说“噪声分布”只是删除像素，因此去噪自动编码器必须在给定噪声版本的情况下重建原始图像。但请注意，这仍然无法解决问题。标准去噪自动编码器仍然要求给定噪声样本的原始样本的条件分布遵循简单的参数分布。对于像图像补丁这样的复杂数据，情况并非如此。

许多[4]，无需我们明确指定输出分布的结构。

给定输入X和输出Y，我们想要创建模型P（YX）这最大化了基本事实的可能性（我为此重新定义X而道歉。但是，标准机器学习符号将X映射到Y，所以我也会这样做）。我们通过引入潜在变量来定义模型

z~N（0，I），这样：

$$P(Y|X) = N(f(z,X), \sigma^2 * I). \tag{12}$$

其中f是确定性函数，我们可以从数据中学习。我们可以重写方程式2 通过5 对X的调节如下：

$$D\left[Q\left(z \mid Y, X\right) P\left(z \mid Y, X\right)\right] = E_{z \sim Q(\cdot|Y,X)}\left[\log Q\left(z \mid Y, X\right) - \log P\left(z \mid Y, X\right)\right] \tag{13}$$

$$D\left[Q(z|Y,X)\ P(z|Y,X)\right] = E_{z \sim Q(\cdot|Y,X)}\left[\log Q\left(z \mid Y, X\right) - \log P\left(Y \mid z, X\right) - \log P\left(z \mid X\right)\right] + \log P\left(Y \mid X\right) \tag{14}$$

$$\log P\left(Y \mid X\right) - D\left[Q\left(z \mid Y, X\right) P\left(z \mid Y, X\right)\right] = E_{z \sim Q(\cdot|Y,X)}\left[\log P\left(Y \mid z, X\right)\right] - D\left[Q\left(z \mid Y, X\right) P\left(z \mid X\right)\right] \tag{15}$$

注意，P（z X）仍然是（0，I），因为我们的模型假设z在测试时独立于X被采样。该模型的结构如图所示6.

在测试时，我们可以通过简单地采样z~N（0，I）从分布P（Y | X）中进行采样。

4 例子

使用Caffe实现这些示例[22]可以在网上找到：
http://github.com/cdoersch/vae_tutorial

4.1 MNIST变分自动编码器

为了演示VAE框架的分布式学习功能，让我们在MNIST上训练一个变分自动编码器。为了表明框架不是很大程度上依赖于初始化或网络结构，我们不使用现有的已发布的VAE网络结构，而是改编基础

[4]通常在机器学习文献中称为“结构化预测”。

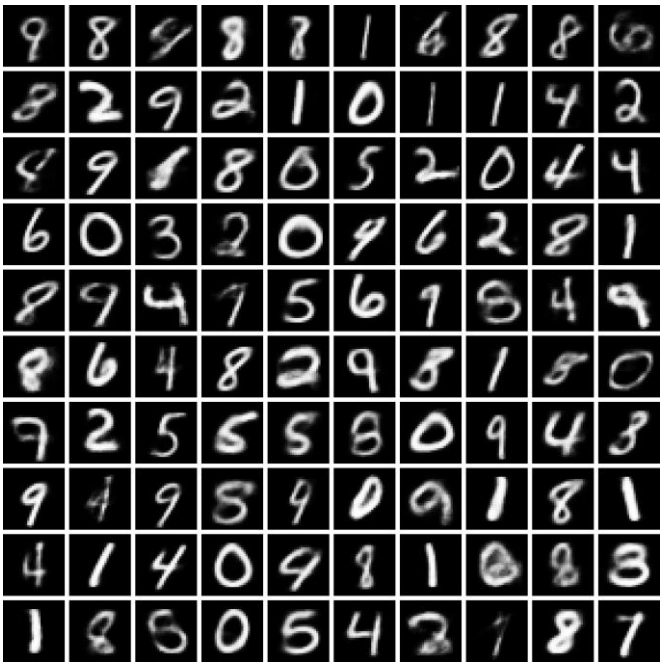
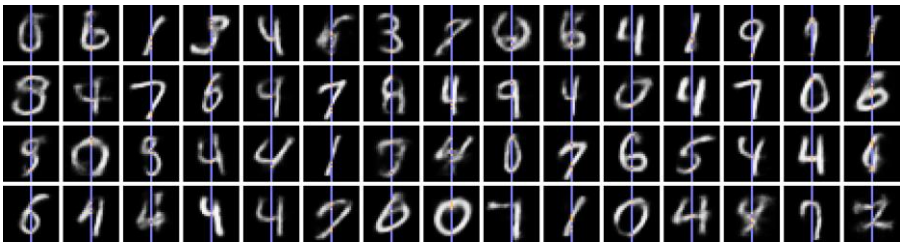
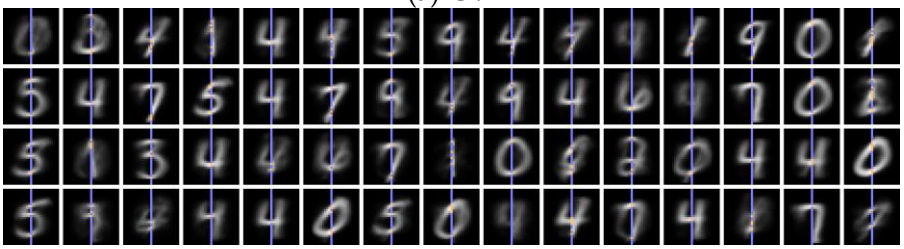


图7：来自在MNIST上训练的VAE的样品。

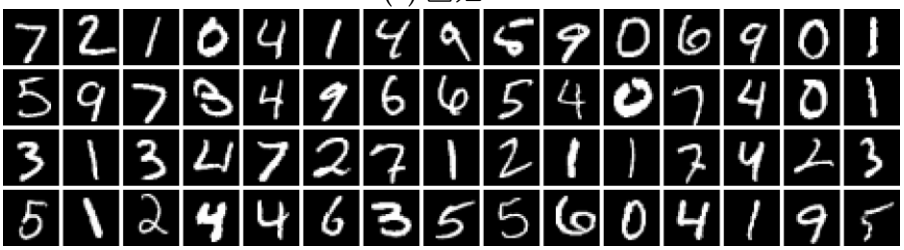
Caffe附带的MNIST AutoEncoder示例[22]. (但是，我们使用ReLU非线性[9]和ADAM [23]，因为两者都是加速收敛的标准技术。) 虽然MNIST是实值的，但它被约束在0和1之间，所以我们使用Sigmoid交叉熵损失
对于P（X | z）。这有一个概率解释：假设我们通过独立地将每个维度采样为$X \sim \text{Bernoulli}(X_i)$来创建新的数据点$X^t$。交叉熵测量X的预期概率^t。因此，我们实际上建模X^t，即MNIST的随机二值化版本，但是
我们只给出了这个数据X的总结。不可否认，这不是VAE框架规定的，但在实践中运作良好，并用于其他VAE文献[6].尽管我们的模型比[[1]和[3]，训练模型并不困难。培训完成了一次完成（尽管重新开始培训5次，以找到使损失下降最快的学习率）。噪声产生的数字如图所示7. 值得注意的是，这些样本很难评估，因为没有简单的方法来衡量这些样本与训练集的差异[24].然而，失败的情况很有趣：虽然大多数数字看起来非常逼真，但很多数字都是“不同的数字”。例如，最左边一列顶部的第七个数字显然位于7和9之间。这是因为我们通过平滑函数映射连续分布。



(a) CVAE



(b) 回归



(c) 地面真相

图8：来自在MNIST上训练的CVAE的样品。模型调节的输入是中心列，在前两个图像中以蓝色和橙色突出显示。该模型必须完成仅给出这些噪声二进制值的数字。上面的三组在空间上对齐，因此您可以将生成的图像与基础事实进行比较。

实际上，除非z过大或过小，否则模型似乎对z的维数非常不敏感。太少的z意味着模型不能再捕获所有变化：少于4个z维度产生明显更差的结果。1,000 z的结果很好，但有10,000个，它们也降级了。理论上，如果具有n z的模型是好的，则具有m >> n的模型不应该更差，因为模型可以简单地学习忽略额外的维度。然而，在实践中，当z非常大时，似乎随机梯度下降努力使[q（z X）P（z）]保持低。

D | ||

5 MNIST条件变分自动编码器

我本来打算显示条件变量自动编码器，只给出每个数字的一半，完成MNIST数字。虽然CVAE可以很好地用于此目的，但不幸的是，回归器实际上也能很好地工作，产生相对清晰的样品。明显的原因是MNIST的规模。与部分容量相似的网络4.1 可以很容易地记住整个数据集，因此回归器严重过度。因此，在测试时，它产生的行为类似于最近邻匹配，这实际上非常尖锐。在给出训练示例的情况下，当输出不明确时，CVAE模型最有可能优于简单回归。因此，让我们对问题进行两次修改，使其更加模糊，但代价是使其更加人为。首先，输入是从数字中间取得的单列像素。在MNIST中，每个像素具有介于0和1之间的值，这意味着即使在该单个像素列中仍存在足够的信息，以便网络识别特定的训练示例。因此，第二个修改是用二进制值（0或1）替换列中的每个像素，选择1的概率等于像素强度。每次将数字传递到网络时，都会重新采样这些二进制值。数字8 显示结果。请注意，回归模型通过模糊其输出来处理模糊性（尽管有些情况下回归量在做出错误猜测时会产生怀疑，这表明过度拟合仍然是一个问题）。回归量输出中的模糊最小化了可能产生输入的许多数字集的距离。另一方面，CVAE通常选择一个特定的数字来输出，并且没有模糊，从而产生更可信的图像。

致谢：感谢UCB CS294视觉对象和活动识别小组和CMU Misc-Read小组中的每个人，以及鼓励我将演示文稿转换为教程的许多其他人。我要特别感谢PhilippKrähenbühl，Jacob Walker和Deepak Pathak帮助我制定和完善我对该方法的描述，并感谢Kenny Marino的帮助编辑。我还要感谢Abhinav Gupta和Alexei Efros的有益讨论和支持，感谢Google支持我的研究。

A在一维证明VAE具有零近似误差给予任意强大的学习者。

设$P_{燃气轮机}$（X）是我们试图使用VAE进行近似的1D分布。我们假设$P_{燃气轮机}$（X）> 0到处都是，它是无限的 -

可怜的，所有的衍生物都是有限的。回想一下，变分自动编码器可以优化

$$\log P_\sigma(X) - D[Q_\sigma(z \mid X) P_\sigma(z \mid X)] \quad (16)$$

其中，对于z~N (0,1)，$P_\sigma(X \mid z) = N(X \mid f(z), \sigma^2)$，$P_\sigma(X) = \int_z P_\sigma(X \mid z) P(z)\, dz$，$Q_\sigma(z \mid X) = N(z \mid \mu_\sigma(X), \Sigma_\sigma(X))$。我们明确依赖σ因为我们将它发送到0以证明收敛。理论上最好的解决方案是$P_\sigma = P_{燃气轮机}$和$[Q_\sigma(z \mid X) P_\sigma(z \mid X)] = 0$.通过“任意强大”的学习者，我们的意思是，如果有存在f，μ和Σ，它们实现了这种最佳解决方案，然后学习算法就能找到它们。因此，我们必须仅表明存在这样的f，μ_σ和Σ_σ。首先，$P_{燃气轮机}$实际上可以任意描述为$P_{燃气轮机}(X) = \int_z N(X \mid f(z), \sigma^2) P(z)\, dz$ as σ接近0.为了表明这一点，让F为累积分布函数$P_{燃气轮机}$的（CDF），并且G是(0,1)的CDF，它们都保证存在。然后G（z）分布为Uni f (0,1)（均匀分布），因此$f(z) = F^{-1}(G(z))$分布为$P_{燃气轮机}(X)$。这意味着当$\sigma \to 0$时，分布P（X）收敛于$P_{燃气轮机}$。

从这里开始，我们必须简单地说明一下$D[Q_\sigma(z \mid X) P_\sigma(z \mid X)] \to 0$为$\sigma \to 0$。设$g(X) = G^{-1}(F(X))$，即f的倒数，设$Q_\sigma(z \mid X) = N(z \mid g(X), (g^t(X))^2 \sigma^2)$.注意$[Q_\sigma(z \mid X) P_\sigma(z \mid X)]$是不变的仿射样本空间的变换。因此，让$Q^0(z \mid X) = N(z \mid g^0(X), (g^t(X))^2)$和$P^0(z \mid X) = P_\sigma(z = g(X) + (z^0 g(X)) \sigma \mid X)$。当我写P（z = ...）时，我使用z的PDF作为函数，并在某些时候进行评估。然后：

$$D[Q_\sigma(z \mid X) P_\sigma(z \mid X)] = D[Q^0(z \mid X) P_\sigma(z \mid X)] \quad (17)$$

其中$Q^0(z^0 \mid X)$不依赖于σ，其标准偏差大于0.因此，足以证明$P^0(z^0 \mid X) \to Q$对于所有z，$^0(z^0 \mid X)$。

设$r = g(X) + (z^0 - g(X)) * \sigma$。然后：

$$\begin{aligned} P^0(z^0 \mid X) &= P_\sigma(z = r \mid X = X) * \sigma \\ &= \frac{P_\sigma(X = X \mid z = r) * P(z = r) * \sigma}{P_\sigma(X = X)} \end{aligned} \quad (18)$$

这里，$P_\sigma(X) \to P_{燃气轮机}(X)$为$\sigma \to 0$，其为常数，rg（X）为σ0，因此P（r）也趋于恒定。与σ一起，它们确保整个分布正常化。我们将它们都包含在常量C中。

$$= C * N(X \mid f(g(X) + (z^0 - g(X)) * \sigma), \sigma^2). \quad (19)$$

接下来，我们围绕g（X）进行f的泰勒展开：

$$= C * \mathcal{N}\left(X \middle| X + f^{t}(g(X)) * (z^{0} - g(X)) * \sigma + \sum_{n=2}^{\infty} \frac{f^{(n)}(g(X))((z^{0} - g(X)) * \sigma)^{n}}{n!}, \sigma^{2}\right) . \tag{20}$$

注意，N（X |μ，σ）=（$\sqrt{2\pi}$σ）$^{-1}$exp $\frac{-(x-\mu)^2}{2}$。我们使用这个公式重写上面的内容，重新排列术语，并将结果重写为高斯，以获得：

$$= \frac{C * f^{t}(g(X))}{\sigma} * \mathcal{N}\left(z^{0} \middle| g(X) - \sum_{n=2}^{\infty} \frac{f^{(n)}(g(X))((z^{0} - g(X)) * \sigma)^{n}}{n! * f^{t}(g(X)) * \sigma}, \frac{1}{f^{t}(g(X))^{2}}\right) . \tag{21}$$

注1 / f ^{t}（g（X））= g^{t}（X），因为f = g^{-1}。此外，由于f $^{(n)}$对所有n都有界，因此总和中的所有项都倾向于0为σ 0。C必须使分布正规化，因此我们得到上述表达式：

$$\rightarrow \mathcal{N}\left(z^{0} | g(X), g^{t}(X)^{2}\right) = Q^{0}(z^{0} | X) \quad \square \tag{22}$$

看方程21，该设置中的大部分近似误差来自g的曲率，其主要由地面实况分布的cdf的曲率决定。

参考

1. Diederik P Kingma和Max Welling。自动编码变分贝叶斯。*i*clr，2014。
2. Tim Salimans，Diederik Kingma和Max Welling。马尔可夫链蒙特卡罗和变分推论：缩小差距。ICML，2015年。
3. Danilo Jimenez Rezende，Shakir Mohamed和Daan Wierstra。深部生成模型中的随机反向传播和近似推断。在ICML，2014年。
4. Tejas D Kulkarni，William F. Whitney，Pushmeet Kohli 和 Josh Tenenbaum。深度卷积逆图形网络。在NIPS，2015年。
5. Diederik P Kingma，Shakir Mohamed，Danilo Jimenez Rezende和Max Welling。具有深层生成模型的半监督学习。在NIPS，2014年。
6. Karol Gregor，Ivo Danihelka，Alex Graves，Danilo Rezende和Daan Wierstra。绘图：用于图像生成的递归神经网络。在ICCV，2015年。

7. Kihyuk Sohn, Honglak Lee和Xinchen Yan。使用深度条件生成模型学习结构化输出表示。在NIPS，2015年。
8. Jacob Walker, Carl Doersch, Abhinav Gupta和Martial Hebert。不确定的未来：使用变分自动编码器从静态图像预测。在ECCV，2016年。
9. Alex Krizhevsky, Ilya Sutskever和Geoff Hinton。深度卷积神经网络的Imagenet分类。在NIPS，2012年。
10. Bruno A Olshausen和David J Field。通过学习自然图像的稀疏代码来出现简单细胞感受野性质。Nature，381（6583）：607-609,1996。
11. Honglak Lee, Alexis Battle, Rajat Raina和Andrew Y Ng。高效的稀疏编码算法。在NIPS，2006年。
12. Pascal Vincent，Hugo Larochelle，Yoshua Bengio和Pierre-Antoine Manzagol。使用去噪自动编码器提取和组合强大的功能。在ICML，2008年。
13. Yoshua Bengio，Eric Thibodeau-Laufer，Guillaume Alain和Jason Yosinski。由backprop训练的深度生成随机网络。ICML，2014年。
14. Yoshua Bengio, Li Yao, Guillaume Alain和Pascal Vincent。广义去噪自动编码器作为生成模型。在NIPS，第899-907页，2013年。
15. L*uc Devroye。基于样本的非均匀随机变量生成。Springer-Verlag，纽约，1986年。*
16. Peter Dayan，Geoffrey E Hinton，Radford M Neal和Richard S Zemel。亥姆霍兹机器。神经计算，7（5）：889-904,1995。
17. Geoffrey E Hinton, Peter Dayan, Brendan J Frey和Radford M Neal。无监督神经网络的"唤醒 - 睡眠"算法。Science，268（5214）：1158-1161,1995。
18. Geoffrey E Hinton, Simon Osindero和Yee-Whye Teh。深度信念网的快速学习算法。神经计算，18（7）：1527-1554,2006。
19. Ruslan Salakhutdinov和Geoffrey E Hinton。Deep boltzmann机器。在人工智能和统计国际会议上，第448-455页，2009年。
20. Geoffrey E Hinton和Drew Van Camp。通过最小化权重的描述长度来保持神经网络的简单性。在1993年第六届计算学习理论年会论文集中。

21. Geoffrey E Hinton和Richard S Zemel。自动编码器，最小描述长度和亥姆霍兹自由能。在NIPS，1994年。

22. Yangqing Jia，Evan Shelhamer，Jeff Donahue，Sergey Karayev，Jonathan Long，Ross Girshick，Sergio Guadarrama和Trevor Darrell。Caffe：用于快速特征嵌入的卷积架构。在ACM-MM，2014年。

23. Diederik Kingma和Jimmy Ba。亚当：随机优化的一种方法。ICLR，2015年。

24. Lucas Theis，Aäronvanden Oord和Matthias Bethge。关于生成模型评估的说明。ICLR，2016年。