【啄米日常】7：Keras示例程序解析（4）：变分编码器VAE

BigMoyan · 7 个月前

今天我们来介绍vae，不是“雨后江岸天破晓，老舟新客知多少”的那位。

VAE，全称variational autoencoder，是一种相对而言比较复杂的深度学习模型。之前的文章都以实现为主，这篇文章除了要把vae的实现讲明白，还想尽量把的vae的原理介绍好。也许会有一些简单的数学公式，但太复杂的推导不会有。变分编码器也是最近才仔细研究的，如有理解不到位的地方，还请各位大神多多拍砖。

通过本文你将学到

- 一点关于自动编码器的说明

- 多路损失函数的配置

主题较难，本鸡战战兢兢，努力把问题讲好，多谢各位捧场。主要的参考文献有两篇，先贴在这以免忘了：

- Auto-Encoding Variational Bayes
- Tutorial on Variational Autoencoders

自动编码器

自动编码器因结构简单，容易上手，一直是刚入门深度学习的同学的最爱。与其说自动编码器属于无监督学习，不如说它属于自监督学习，它还是有标签的，只不过标签就是自己，不需要人类网上打而已。

自动编码器种类繁多，有栈式自动编码器，去噪自动编码器，稀疏自动编码器，卷积自动编码器等等。总的来讲，算是深度学习里一个小门类。自动编码器最早是用于深度神经网络的预训练，使早期的比较深的神经网络能够顺利训练出来。随着relu和新的权重初始化方法和网络结构的提出——主要是ReLU的应用，训练深度神经网络再也不用自动编码器预训练了。

所以私信我问栈式自动编码器怎么写的同学们，在这里统一回复，这个玩意我真写过，就是一层一层的预训练，上一层的结果作为下一层的输入。但是我不告诉你怎么写，也劝你别研究，自动编码器本身就不是一个特别有研究点的东西，栈式的预训练更是早就淘汰的技术，只不过网上的一些教程还没更新而已。Keras 0.x版本还有AutoEncoder这个层，后来直接都删了。

变分编码器是一种比较另类的自动编码器，或者说，压根就不是自动编码器。

隐变量

考虑MNIST数据集，数据集里有10种数字，每种数字下有几千个不同的样本，我们能不能照猫画虎，模仿已有的数字生成一个同样可辨识，但却与现有的样本都不同的的数字呢？

要解决这个问题，需要对数字的分布进行建模。我们需要知道一个数字“一般而言长什么样”。如果我们得到了数据集X的分布P(X)，那么数据集中的每个图片，也不过就是从P(X)采样得到的一个样本而已。可以说掌握了P(X)，我们就算把这个数据集的底裤都扒下来了，到时候搓扁捏圆，任君所愿。

然而从有限的样本中估计出数据原来的分布情况，却不是一件容易的事，别说更复杂的数据集

一组我们观察不到的隐变量Z经过某个复杂的映射 $f(z;\theta)$ 产生的，给定一个z，我就能通过某种方法得到一个样本x。如果能得到z的分布和 $f(z;\theta)$ ，那P(X)我们也算知道了。

这个假设是有道理的，z尽管是隐变量，但不妨碍我们对它的物理含义做出猜测。比方说z里面有控制笔画粗细的变量，有控制笔划角度的变量，有控制数字大小的变量等等。这些变量一旦确定，写出来的数字大致长什么样就确定了。

为什么要多此一举呢？由原来的估计P(X)变成估计P(Z)跟 $f(z;\theta)$ 两个东西，还麻烦了一层。主要的理由有两点：

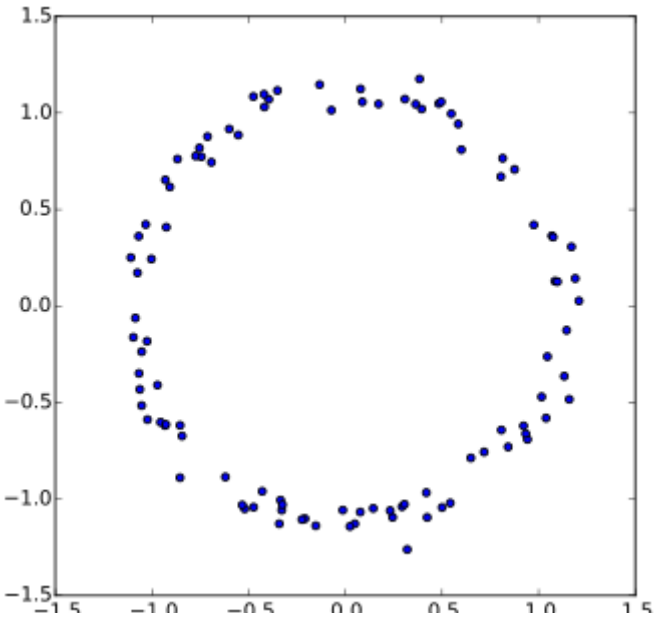
- 估计P(Z)能避免直接估计P(X)的一个大麻烦，你无法知道P(X)分布的形式，但我们完全可以假定P(Z)服从标准n维高斯分布。
- $f(z;\theta)$ 虽然可能是个很复杂很复杂的映射，但这映射不需要你手动设计，可以通过学习得到

第一点看起来非常不可思议，P(X)分布的形式未知这点很好理解，但凭什么P(Z)能服从标准n维高斯分布呢？比方z里面有个控制比划粗细的变量，它怎么可能是高斯分布呢？

理由要落在 $f(z;\theta)$ 的复杂性上， 不知道下面这句话大家是否能够认同：

对于任意d维随机变量，不管他们实际上服从什么分布，我总可以用d个服从标准高斯分布的随机变量通过一个足够复杂的函数去逼近它。

举个例子，假如我们要通过一个2维联合高斯分布的z去逼近一个在二维平面上呈环状分布的分布，则函数 $g(z)=z/10+z/||z||$ 可以做到，经过g映射的分布长这样：



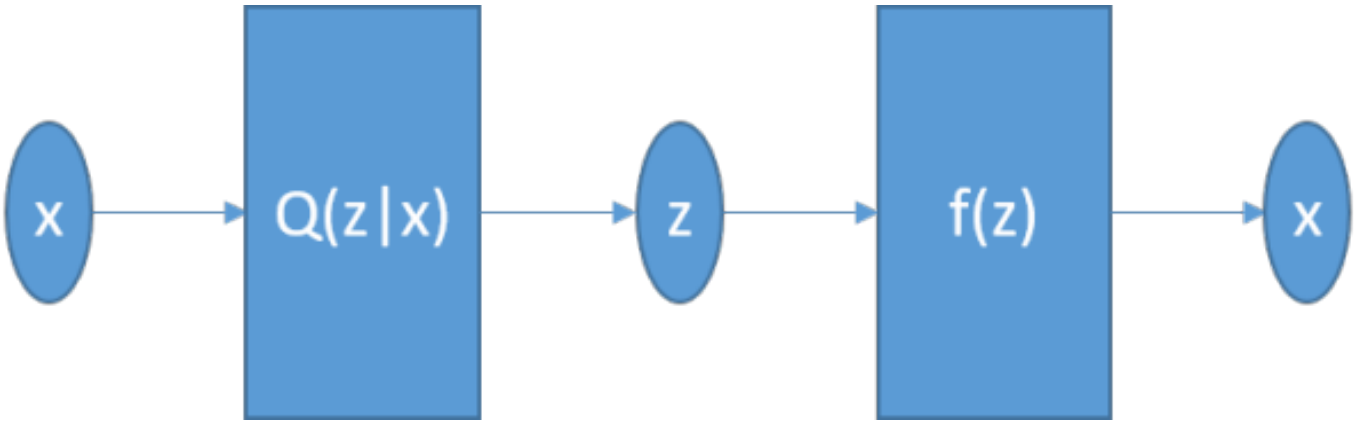
z，这个z可能就是上面说的笔划粗细嘛，用度什么的有物理含义的东西，然后接着把z映射到X得到样本。

当然，这是为了方便理解给出的解释，实际上未必是这样。总而言之，可以假定Z服从n维高斯分布的理由，只要 $f(z;\theta)$ 足够复杂。而对数据集分布的建模，就落在对复杂映射函数 $f(z;\theta)$ 的构建上，在变分编码器里，这个复杂函数就由多层网络来实现。

变分编码器

既然z是服从标准正态分布的，那是否可以通过不断从P(z)里采样来训练 $f(z;\theta)$ 呢？这是不行的，理由之一是你不知道该对采样出来的这个样本指定哪张图片作为它的标签。要知道现在我们 $f(z;\theta)$ 还是一团混沌，还没有真正的开始训练。也就是说，虽然z是服从标准正态分布的，但在训练阶段，z的具体值还要依靠训练样本x来出。

由x所产生的z的样本，最能够反应在x条件下z应该是什么样子，也最能指导重建网络如何从z中重构回x。从x中产生z的方法，我们叫做Q(z|x)。依据Q(z|x)产生的z，一方面，我们要求它服从标准正态分布。另一方面，我们还要求 $f(z;\theta)$ 能够将z还原为x。Q(z|x)就成为了联系数据集样本x与隐变量z的纽带。从x到z的这个过程，我们仍然还用一个MLP来实现。到这一步我们的网络是这样的（以下各图圆圈代表数据，方框代表过程）：



恭喜你获得了一只新鲜出炉的**自动编码器**

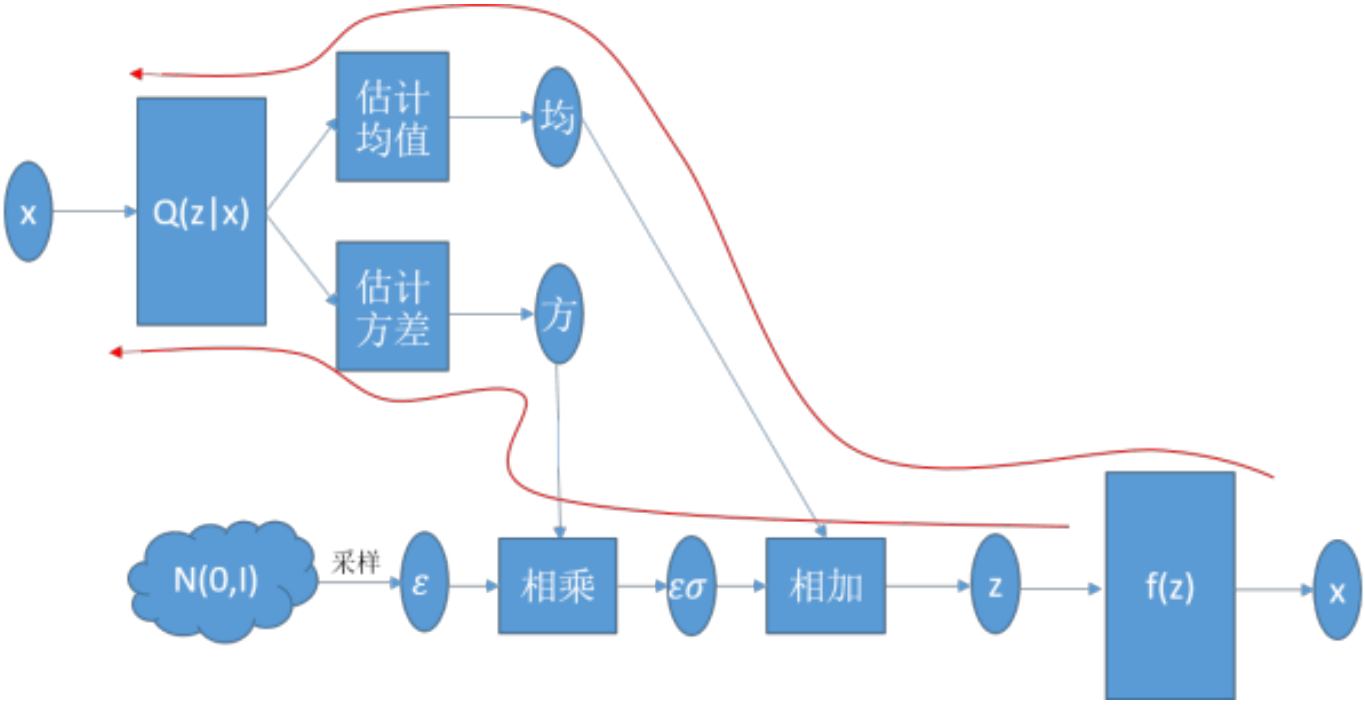
说好的随机变量呢，说好的正态分布呢！说好的抽出一个样本z就能产生一个x呢！

变分编码器和自动编码器的区别就在于，传统自动编码器的隐变量z的分布是不知道的，因此我们无法采样得到新的z，也就无法通过解码器得到新的x。下面我们来变分，我们现在不要从x中直接得到z，而是得到z的均值和方差，然后再迫使它逼近正态分布的均值和方差，则网络变成下面的样子：

看上去不错，从Q(z|x)估计出来的值跟标准正态分布不一样没关系，训练过程中慢慢逼近就行了。假定z服从高斯分布的好处之一在这里就能体现出来，只要估计均值和方差，我们就完全了解这个高斯分布了，也就能从其中采样了。

然而上面这个网络最大的问题是，它是**断开**的。前半截是从数据集估计z的分布，后半截是从一个z的样本重构输入，最关键的采样这一步，恰好不是一个我们传统意义上的操作，这个网络没法求

为了使得整个网络得以训练，使用一种叫**reparemerization**的trick，使得网络对均值和方差可导，把网络连起来。这个trick的idea见下图：



实际上，这是将原来的单输入模型改为二输入模型了。因为 ε 服从标准正态分布，所以它乘以估计的方差加上估计的均值，效果跟上上图直接从高斯分布里抽样本结果是一样的。这样，梯度就可以通上图红线的方向回传，整个网络就变的可训练了。

目标函数和测试

目标函数有两项，一项与自动编码器相同，要求从f(z)出来的样本重构原来的输入样本。另一项我们之前提到了，要求经过Q(z|x)估计出的隐变量分布接近于标准正态分布。前一项由重构x与输入x均方差或逐点的交叉熵衡量，后一项衡量两个分布的相似度，当然是大名鼎鼎的KL距离。

网络的梯度通路打通后，网络的训练与普通自动编码器没有什么不同。但是由于VAE的特殊性，使用的时候跟普通自动编码器不一样。在使用的时候我们需要将编码器端全部抛掉，之前说了，对数据集的建模就体现在复杂函数 $f(z;\theta)$ 上，搞这一堆又是Q又是reparemerization的，说到底都是为了训练 $f(z;\theta)$。一旦 $f(z;\theta)$ 训练完毕，我们只需要从高斯分布中采样，然后送进去 $f(z;\theta)$，就可以生成新样本x了。

Keras实现

下面我们来看Keras对vae的实现，这里的例子是mnist数据库

```
intermediate_dim = 256
nb_epoch = 50
epsilon_std = 1.0

x = Input(batch_shape=(batch_size, original_dim))#1
h = Dense(intermediate_dim, activation='relu')(x)#2
z_mean = Dense(latent_dim)(h)
z_log_var = Dense(latent_dim)(h)
```

首先来构造Q(z|x)，mnist图片的大小是28*28，所以输入维度是784维。#1声明了一个给定形状的张量占位符，该占位符可以被随后的层映射。#2 是一个将784映射到256的全连阶层，然后z_mean和z_log_var分别是估计出来的均值和方差。

根据框图，这时候我们需要一个层来产生样本。产生的原理是生成一个高斯分布的样本，然后跟z_mean和z_log_var搅合起来。因此这个层是一个二输入单输出的层。因为它只是完成一个简单的张量运算动作而不涉及可训练参数，用Lambda层正合适。

Keras的Lambda层以一个张量函数为参数，对输入的数据按照张量函数的要求做映射。本质上就是Keras layer中.call()的快捷方式。先定义运算逻辑：

```
def sampling(args):
    z_mean, z_log_var = args
    epsilon = K.random_normal(shape=(batch_size, latent_dim), mean=0.,
                              std=epsilon_std)
return z_mean + K.exp(z_log_var / 2) * epsilon
```

注意这里函数期望输入是一个由两个参数构成的tuple或者list或者别的可以unpack的东西，并期望args的第一个参数代表均值，第二个代表方差。epsilon是从标准正态分布抽出来的样本，其shape是（batch_size, latent_dim），因为网络对样本是按照batch处理的

随后，在Lambda的包装下，我们顺利获得了采样到的z样本：

```
z = Lambda(sampling, output_shape=(latent_dim,))([z_mean, z_log_var])
```

然后我们搭建解码端，同样用一个MLP作为解码器，将z解码到786维向量：

```
decoder_h = Dense(intermediate_dim, activation='relu')
decoder_mean = Dense(original_dim, activation='sigmoid')
h_decoded = decoder_h(z)
x_decoded_mean = decoder_mean(h_decoded)
```

网络这就算搭完了，然后准备训练。我们需要配置目标函数，目标函数由两项。按照参考文献1，重构项的损失用逐点的logloss表示，估计出来的分布与标准正态分布的误差用KL距离表示：

```
def vae_loss(x, x_decoded_mean):
    xent_loss = original_dim * objectives.binary_crossentropy(x, x_decoded_mean)
    kl_loss = - 0.5 * K.sum(1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-
return xent_loss + kl_loss
```

xent就是x_entropy的简写，因为logloss的另一个名字就是二类交叉熵。kl距离这里是按照参考文献手写的。具体推导略过。注意这里xent_loss这项调用了Keras自己的损失函数binary_crpssentropy，Keras的损失函数是求了均值的，这里乘了一个维度又scale回去了。

然后是常规的训练，不必多说

```
vae = Model(x, x_decoded_mean)
vae.compile(optimizer='rmsprop', loss=vae_loss)

# train the VAE on MNIST digits
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))

vae.fit(x_train, x_train,
        shuffle=True,
        nb_epoch=nb_epoch,
        batch_size=batch_size,
validation_data=(x_test, x_test))
```

最后，这份代码还提供了使用vae的一个例子，正如之前所说，剪掉编码器部分，直接把正态分布样本送入解码器即可

```
decoder_input = Input(shape=(latent_dim,))
_h_decoded = decoder_h(decoder_input)
_x_decoded_mean = decoder_mean(_h_decoded)
generator = Model(decoder_input, _x_decoded_mean)
```

结束语

下次更啥呢……

管他呢，又可以两周不更新了！喵喵喵！

「帮菜鸡凑钱GTX1080吧！」

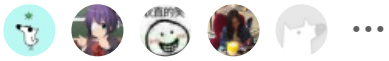
赞赏

5 人赞赏

深度学习（Deep Learning） keras

☆ 收藏 分享 举报

82



18 条评论

写下你的评论...

SCP-173

首赞才是真爱

7 个月前 1 赞

Shawn

感谢群主

知 首发于 **菜鸡的啄米日常**

写文章 登录

pires

谢谢示例，写的非常通俗易懂。我有一个小问题，z的维度是2的原因是图片输入维度也是2吗？不知道升维和降维对结果有什么影响？

7 个月前 1 赞

柴云

赞

7 个月前

BigMoyan（作者） 回复 pires 查看对话

z的维度也可以是其他的，你可以随便设一下试试。

7 个月前

Lee的海

我记得有人说VAE的一个限制条件是它要求一个什么分布要能够factorize（P(AB..)=P(A)P(B)..这样），不知道什么为什么来着..？

7 个月前

BigMoyan（作者） 回复 Lee的海 查看对话

你写的这不是独立吗？

7 个月前

艾柯斯

autoencoder 翻译成自编码器更合适一点吧 毕竟这里auto并不是取“自动的”义而是“自身的”

7 个月前

BigMoyan（作者） 回复 艾柯斯 查看对话

auto似乎没有“自身的”这个意思...就是自动编码器，指的是编码的过程是自动的

7 个月前

ling wei

下一页

文章被以下专栏收录

菜鸡的啄米日常

别拿菜鸡不当鸡！（啥？）

进入专栏

推荐阅读

【消息】Keras 2参上！这次也请大家多多关照哦

昨天，有人在群里po了一条消息：果然……今天早晨起床铺天盖地都是Keras 2发布的消息，吓得… 查看全文 >

BigMoyan · 6 个月前 · 发表于 菜鸡的啄米日常

【消息】我没死，以及一些消息

Hi……大家好哟！最近一段时间专栏得到了许多人的关注，我诚惶诚恐，因为我这更新速度……实… 查看全文 >

BigMoyan · 7 个月前 · 发表于 菜鸡的啄米日常

劳务派遣：不得不说的秘密（二）——派遣的撤回及解除

劳务派遣单位与员工签订的劳动合同中有哪些风险？劳务派遣单位可否主动将劳动者召回？用人单位在何种情况下可以将员工退回？劳务派遣中，竞业限制和服务期如何适用？一、劳务派遣单位 经营… 查看全文 >

洪文律所 · 21 小时前 · 编辑精选

美国互联网创业者为什么支持民主党？一份新的调查数据

如果创业者只考虑个人经济状况，他们应该都坚决拥护减税的共和党。意识形态方