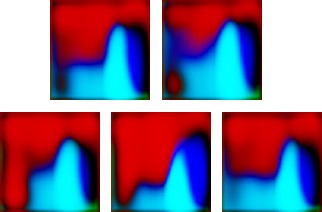
本期继续关注GANs，上期我们也提到，GANs很难训练，也容易发现模式坍塌（产生的结果单一，sharp，不稳定）。的确，上期推送以后，我自己尝试实现GAN，拿STL10数据集作为训练集，发现训练的时候loss一直跳来跳去，或许这是因为generator和discriminator在互博？而产生的图像也不好，下图是迭代1000次以后产生的5张图像，看效果应该是没有训练好（产生的图像很奇怪、模式坍缩也比较严重），然而学习率、训练方法做了一些更换，还是没得到更好的结果。这是个很奇怪的现象，我会继续关注这个问题。



GANs难训练、稳定性差，怎么办？Alec Radford在论文中提出了一些规则，用以提升GAN的稳定性。虽然没有理论上的保证，但是应用了相关规则的DCGAN在不同数据集上的效果都异常地好。好到什么程度呢？训练好的generator能够将相似的输入映射成相似的图像，也就是说，generator能够很好地保留输入的语义相关性。而discriminator能够用于提取通用有效的图像特征，可以用于分类任务。

下面我们来看一下DCGAN的论文[1]都做了些什么（以下用DCGAN代表文献[1]）。

**DCGAN的贡献**

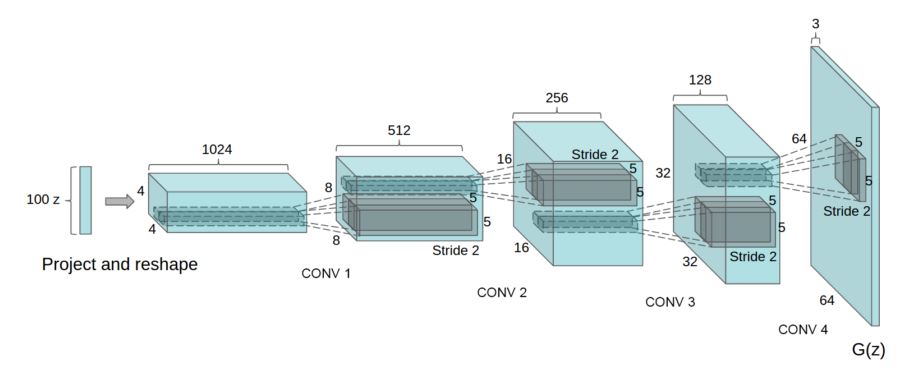
* 提出了一类基于卷积神经网络的GANs，称为DCGAN，它在多数情况下训练是稳定的。
* 与其他非监督方法相比，DCGAN的discriminator提取到的图像特征更有效，更适合用于图像分类任务。
* 通过训练，DCGAN能学到有意义的 filters。
* DCGAN的generator能够保持latent space到image的“连续性”。

**DCGAN model**

实际上，DCGAN是一类GAN的简称，满足以下设计要求（这些要求更像是一些tricks）的GAN网络都可以称为DCGAN模型。

* 采用全卷积神经网络。不使用空间池化，取而代之使用带步长的卷积层（strided convolution）。这么做能让网络自己学习更合适的空间下采样方法。PS：对于generator来说，要做上采样，采用的是分数步长的卷积（fractionally-strided convolution）；对于discriminator来说，一般采用整数步长的卷积。
* 避免在卷积层之后使用全连接层。全连接层虽然增加了模型的稳定性，但也减缓了收敛速度。一般来说，generator的输入（噪声）采用均匀分布；discriminator的最后一个卷积层一般先摊平（flatten），然后接一个单节点的softmax。
* 除了generator的输出层和discriminator的输入层以外，其他层都是采用batch normalization。Batch normalization能确保每个节点的输入都是均值为0，方差为1。即使是初始化很差，也能保证网络中有足够强的梯度。
* 对于generator，输出层的激活函数采用Tanh，其它层的激活函数采用ReLU。对于discriminator，激活函数采用leaky ReLU。

下图是DCGAN的generator model的一个示例，作者用它在LSUN数据集上训练。



我发现之前写的GAN的generator跟这个几乎是一样的（generator的model是在github上找的），除了上采样和卷积的stride。没得到好的效果，大概我用的是一个假模型？

**实验**

作者给了非常详细的实验配置信息：

**|** Batch size: 128;

**|** Learning rate: 0.0002

**|** Leak of leaky ReLU: 0.2

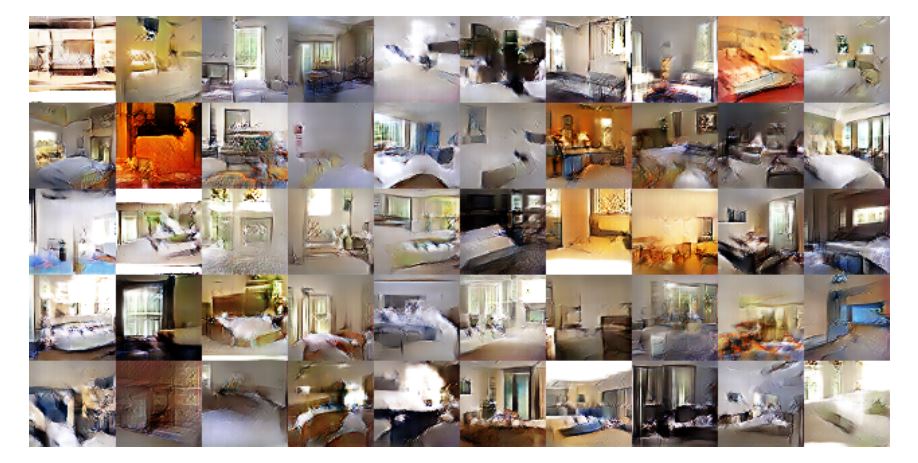
**|** 输入线性映射到[-1, 1]

**|** 所有权重用均值为0，方差为0.02的正态分布随机初始化

**|** 训练方法采用Adam，其中beta\_1 = 0.5

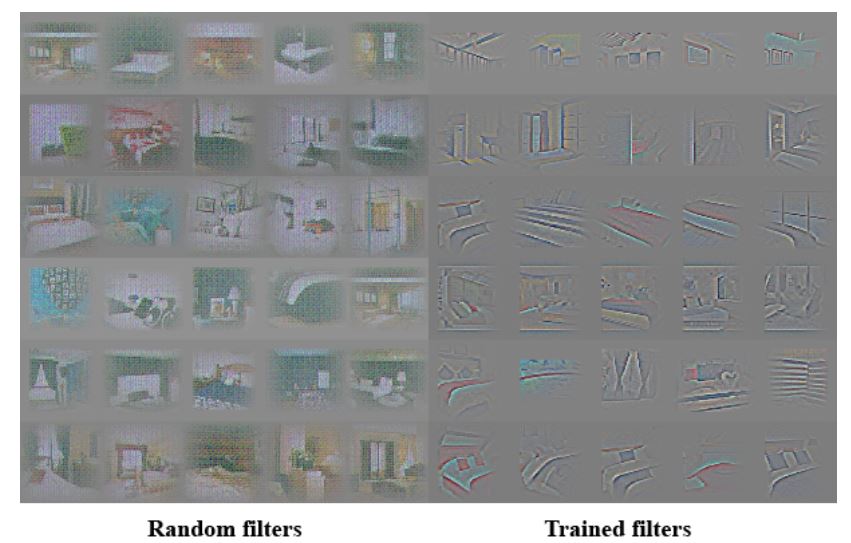
* **生成图像**

下图是上面介绍的generator产生的bedrooms，可以看出来，效果很好，当然，噪声还是存在的。



* **Filter可视化**

下图是discriminator的filters可视化，在大数据集上训练能得到一些有意思的层次特征（hierarchical features）。



* **Generator的“连续性”**

用DCGAN训练的generator能够保持从latent space到image space的“连续性”，这里用了“连续性”，并不是说满足严格的分析上的连续性，而是在latent space上做一点小变化，不会引起image space的大变化。我们不妨称latent vector为generator产生的图像的“图向量”，上面的结论就是说，相似“图向量”通过generator会产生相似的图像。

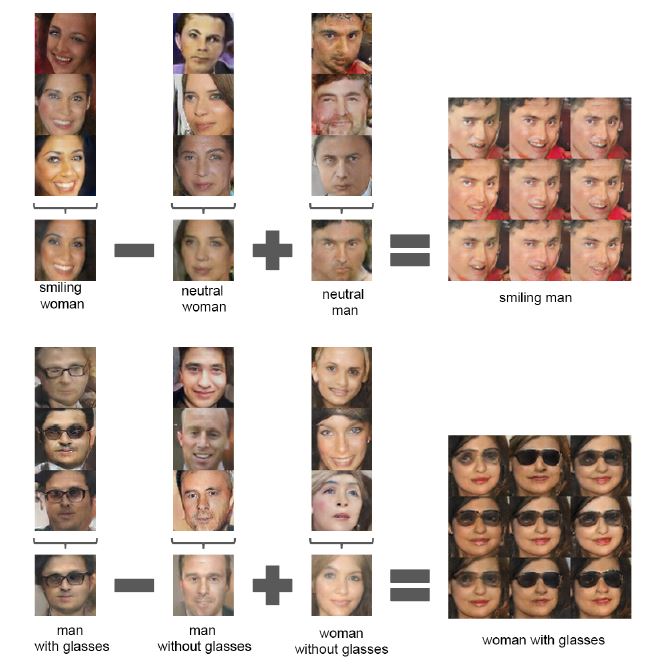
作者在两个不同的“图向量”之间插值得到新的“图向量”（类比于文本的“词向量”），发现它们对应的图像具有一个平滑变换的过程，参见下图所示：





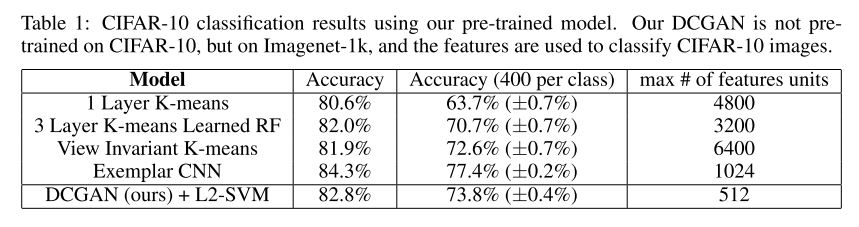
* **Generator能保留“图向量”的语义信息**

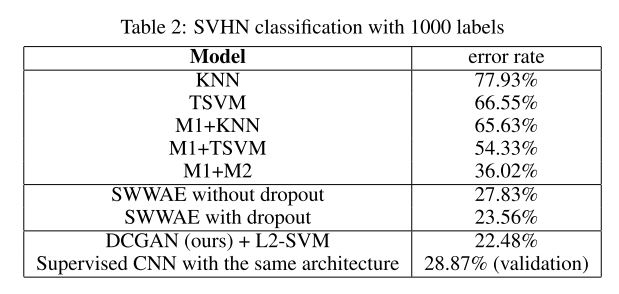
我们知道“词向量”能够比较好地完成类比的任务（不知道的话就看另外一篇推送文章word2vec吧），既然generator能够将“图向量”解码为图像，那它能不能做类比？什么叫图像类比呢？举个例子，戴眼镜的男人 vs 不带眼镜的男人 = 戴眼镜的女人 vs 不带眼镜的女人，也就是说，图像的内容可以做类比。DCGAN训练的generator效果很好，能够比较有效地保留“图向量”的语义信息。下图在图像上的“运算”对应着它们的“图向量”的运算。



* **图像分类**

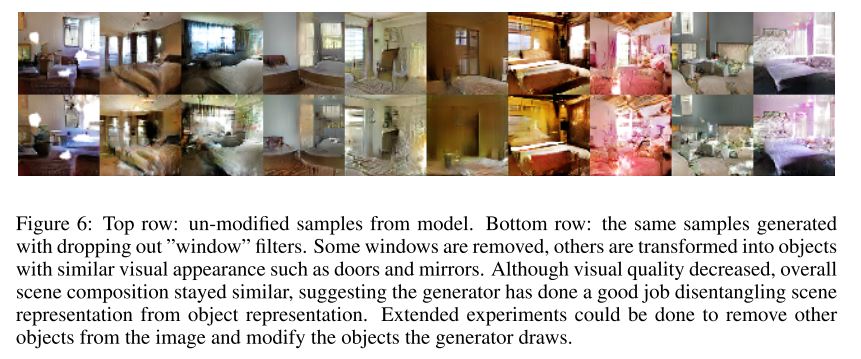
Discriminator能够用于提取图像的特征，作者测试了用DCGAN非监督提取的图像特征（PS：不是直接拿最后一个卷积层的输出，而是每个卷积层的输出做max pooling得到很多4\*4的特征，然后把它们拼接作为图像的特征），训练分类器用于图像分类，得到了不错的效果，也就是说，DCGAN能够提取图像比较通用且有效的特征。实验效果如下图所示





* **其他实验**

作者为了测试DCGAN的效果，还提出了object dropout，用generator的最后一个卷积层的输出训练一个softmax，区分输出图像是否含有特定的object，然后softmax的系数大于0的位置所对应的卷积层的输出都置为0。实验发现这样产生的图像会“忘记”产生特定的object。



**代码**

如果觉得自己动手写DCGAN有困难，不妨先看看别人怎么写的，把它弄懂，然后自己重头写一遍。点击阅读原文可以查看DCGAN的作者提供的代码。

1. DCGAN作者提供的代码（Theano版本，通俗易懂）： <https://github.com/Newmu/dcgan_code>
2. Tensorflow版本： <https://github.com/carpedm20/DCGAN-tensorflow>
3. Torch版本： <https://github.com/soumith/dcgan.torch>

**参考文献**

1. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

**预告**

下期我们继续研究GANs，初步计划是研究InfoGAN，以及GANs的更有效的训练方法，参考文献：

1. Chen X, Duan Y, Houthooft R, et al. Infogan: Interpretable representation learning by information maximizing generative adversarial nets[C]//Advances in Neural Information Processing Systems. 2016: 2172-2180.
2. Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training gans[C]//Advances in Neural Information Processing Systems. 2016: 2226-2234.