

DCGAN

Unsupervised representation learning with deep convolution generative adversarial networks

notes written by h1astro

将CNN网络应用到GAN上，这个idea应该很容易想到。本文作者能脱颖而出的优点在于：代码基本功很扎实，调参能力有，CV和NLP双修（重要改进点）、还考虑了latent space。

用于无监督表征学习的深度卷机生成式对抗网络

核心要点

1. 希望能让CNN在无监督学习上，达到与监督学习一样的成功
2. 通过架构约束，构建了深度卷机生成对抗网络（DCGAN）
3. 证明了DCGAN是目前先进的无监督学习网络
4. 证明了DCGAN的生成器和判别器学习到了从物体细节到整体场景的多层次表征
5. 证明了DCGAN判别器提取的图像特征具有很好的泛化性。

研究背景

表征学习

- 表征 (representation)、特征 (feature)、编码 (code)
- 好的表征
 - 具有很强的表示能力，即同样大小的向量可以表示更多信息
 - 使后续的学习任务变得简单，即需要包含更高层的语义信息
 - 具有泛化性，可以应用到不同领域
- 表征学习的方式
 - 无监督表征学习
 - 有监督表征学习



模型可解释性

-- Interpretation is the process of giving explanations to Human

- 决策树就是一个具有良好可解释性的模型
- 使用特征可视化方法
- 使用数据分析，找到数据中一些具有代表性和不代表性的样本
- NIPS 2017会议上，Yann LeCun：人类大脑是非常有限的，我们没有那么多脑容量去研究所有东西的可解释性。

数据集

LSUN (Large-scale Scene Understanding)

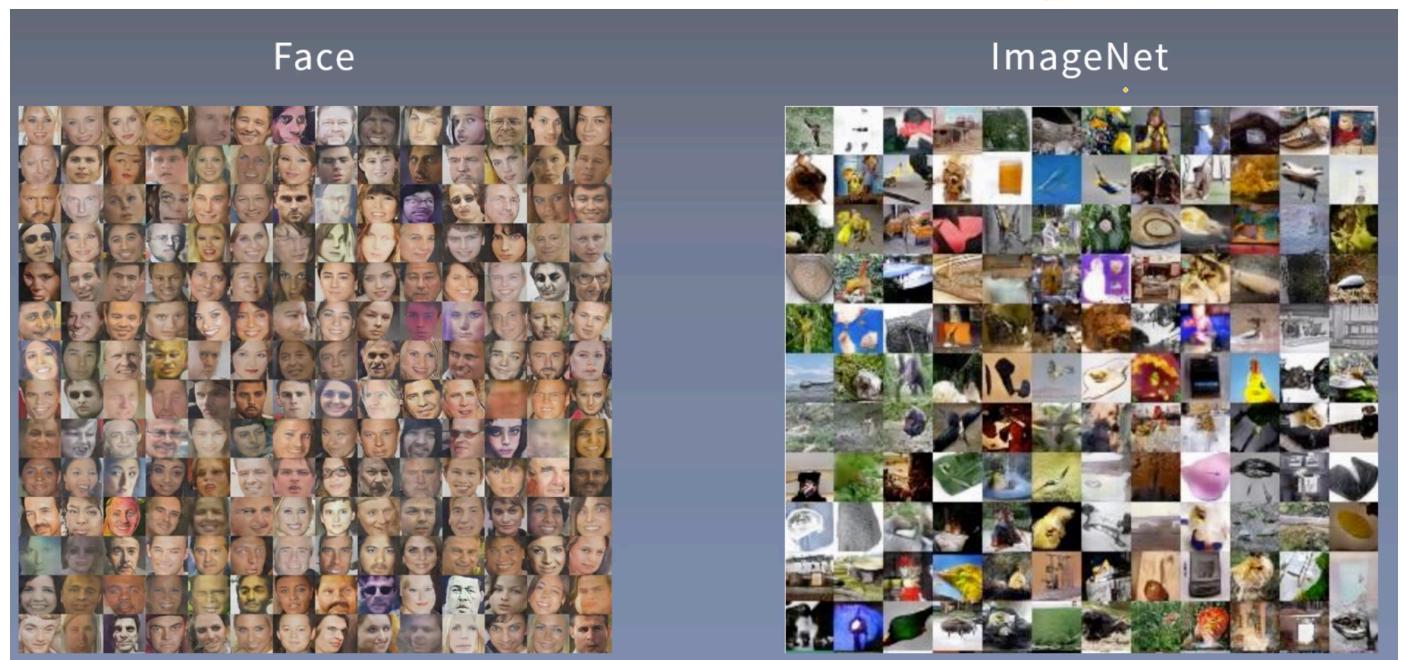
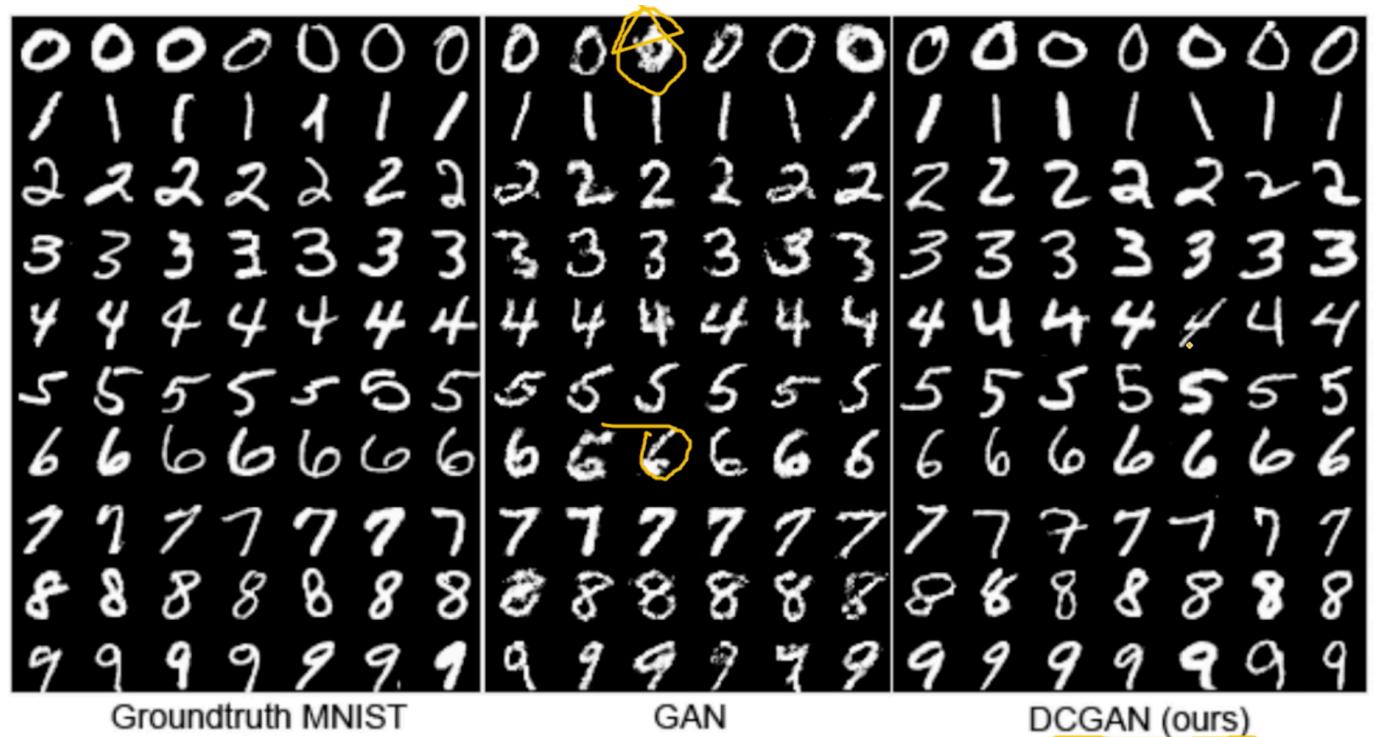
- 加州大学伯克利分校发布，包含10个场景类别和20个对象类别，主要包含了卧室、客厅、教室等场景图像，共计约100万张标记图像

lsun.cs.princeton.edu/2017

SVHN(Street View House Numbers)

- 街景门牌号码数据集，与MNIST数据集类似，但具有更多标签数据（超过600,000个图像），从谷歌街景中收集得到

ufldl.stanford.edu/housenumbers

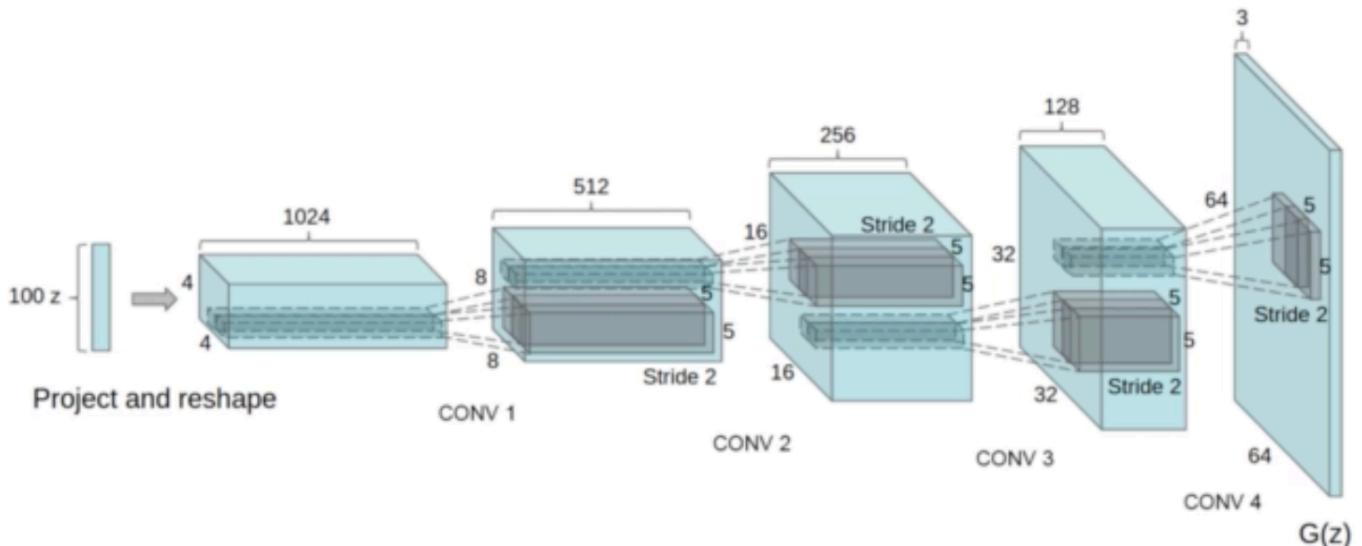


研究意义

- 早起的GAN在图像上仅局限MNIST这样简单数据集中，DCGAN使GAN在图像生成任务上的效果大大提升
- DCGAN几乎奠定了GAN的标准架构，之后GAN的研究者们不用再过多关注模型架构和稳定性，可以把更多的精力放在任务本身上，从而促进了GAN在16年的蓬勃发展
- 开创了GAN在图像编辑上的应用

模型结构

- 所有的pooling层使用strided卷积（判别器）和fractional-strided卷积（生成器）进行替换
- 使用batch normalization（收敛速度、泛化性+）
- 移除全连接的隐层，让网络可以更深
- 在生成器上，除了输出层使用Tanh外，其它所有层的激活函数都使用ReLU
- 判别器所有层的激活函数都使用LeakyReLU



训练参数

- 训练图像预处理，只做了【-1, 1】的阈值缩放
- 使用mini-batch随机梯度下降，batch size 128
- 采用均值为0 标准差为0.02的正太分布，对所有权重进行初始化
- 对于LeakyReLU激活函数，leak斜率设置为0.2
- 优化器使用Adam，不是此前GAN网络的momentum
- Adam的学习速率使用0.0002，而非原论文建议的0.001
- Adam的参数momentum term β_1 ，原论文建议的0.9会导致训练震荡和不稳定，减少至0.5可以让训练更加稳定

LSUN

- 没有使用Data Augmentation
- 在LSUN上训练一个3072-128-3072的自编码器，用它从图像中提取128维特征，再经过ReLU层激活后作为图像的语义hash值
- 对生成图像和训练集使用上面的编码器，提取128维的语义hash值，进行重复性检测
- 检测到了约27.5w左右数量的重复数据（LSUN数据集大小为300多万）

FACES

- 从DBpedia上获取人名，并保证他们都是当代人
- 用这些人名在网络上搜索，收集其中包含人脸的图像，约1w人的300w张图像
- 使用OpenCV的人脸检测算法，截取筛选出较高分辨率的人脸，最终得到了大约35万张人脸图像
- 训练时没有使用Data Augmentation

01 模型结构

网络结构设计

02 图像生成

超参数设置，实验结果

03 无监督表征学习

将表征用于图像分类

04 模型可视化

判别器的最后一层卷积

05 隐空间分析

Walking，去除物体，矢量运算

06 总结展望

展望未来改进方向

07 论文总结

总结论文中创新点、关键点及启发点

无监督表征学习

CIFAR-10

- 在Imagenet-1k上训练DCGAN
- 使用判别器所有层的卷积特征，分别经过最大池化层，在每一层上得到一个空间尺寸为4*4的特征，再把这些特征做 flattened和concatenated，最终得到28672维的向量表示
- 用一个SVM分类器，基于这些特征向量和类别label进行有监督训练

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% ($\pm 0.7\%$)	4800
3 Layer K-means Learned RF	82.0%	70.7% ($\pm 0.7\%$)	3200
View Invariant K-means	81.9%	72.6% ($\pm 0.7\%$)	6400
Exemplar CNN	84.3%	77.4% ($\pm 0.2\%$)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% ($\pm 0.4\%$)	512

SVHN(Street View House Numbers)

- 使用与CIFAR-10实验相同的处理流程
- 使用1w个样本作为验证集，将其用在超参数和模型的选择上
- 随机选择1k个类别均衡的样本，用来训练正则化先行L2-SVM分类器
- 使用相同的生成器结构、相同的数据集，从头训练有监督CNN模型，并使用验证集进行超参数搜索

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWVAE without dropout	27.83%
SWVAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

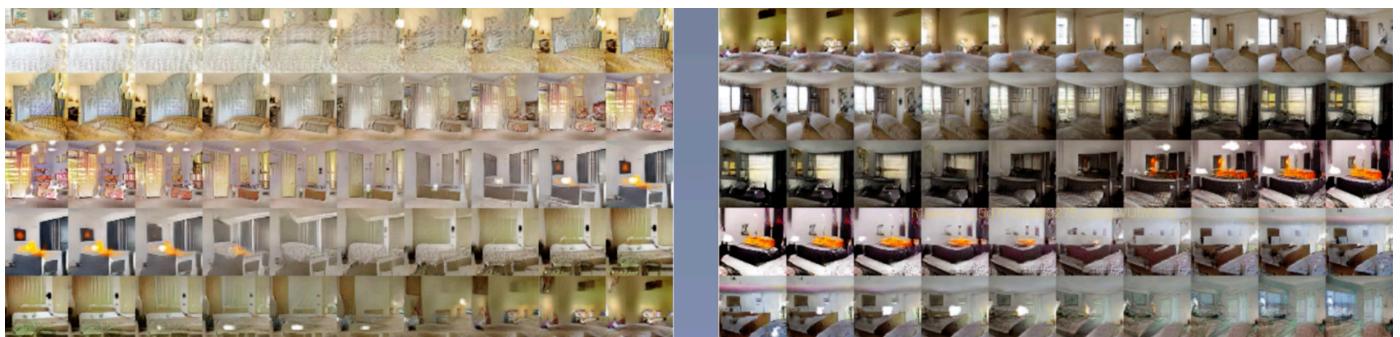
模型可视化

- 在大型图像数据集上训练的有监督CNN模型，可以提取很好的图像feature
- 希望在大型图像数据集上训练的无监督模型DCGAN，也能学习到不错
- 使用对判别器的最后一个卷积层使用特征可视化
- 判别器学习到了卧室的典型部分，例如床和窗户
- 使用随机初始化还未经训练的模型来作为对照

隐空间分析

隐变量空间漫游

- latent space上walking，可以判断出模型是否是单纯在记住输入（如果生成图像过度非常sharp），以及模型崩溃的方式
- 如果在latent space中walking导致生成图像的语义变化（例如添加或删除了对象），可以推断模型已经学习到了相关和有趣的表征



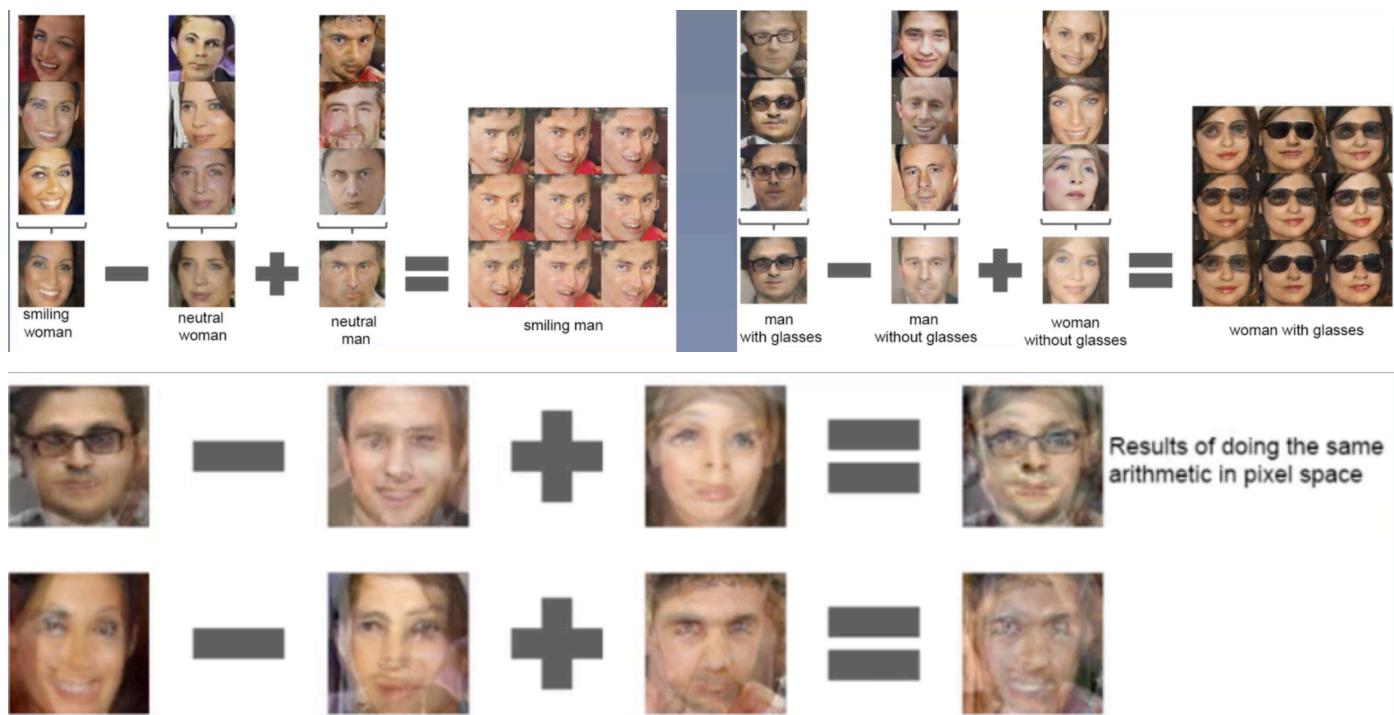
去除特定的对象

- 研究模型是如何对图像中的特定物体进行表征，尝试从生成图像中把窗口进行一取出
- 选出150个样本，手动标注52个窗口的bounding box
- 在倒数第二层的conv layer features中，训练一个简单的逻辑回归模型，判断一个feature activation是否在窗口中
- 使用这个模型，将所有值大于0的特征（总共200个），都从空间位置中移除



人脸样本上的矢量运算

- $\text{Vector}(\text{"King"}) - \text{vector}(\text{"Man"}) + \text{vector}(\text{"Woman"})$ 的结果和向量 Queen 很接近
- 对单个样本进行操作的结果不是很稳定，而如果使用三个样本的平均值，结果就会好很多





总结展望

- 提出了一套更稳定的架构来训练生成对抗性网络
- 展示了对抗性网络可以很好的学习到图像的表征，并使用在监督学习和生成式的建模上
- 模式崩溃问题仍然存在
- 可以再延伸应用到其它领域，例如视频（做帧级的预测）和声频（用于语音合成的与训练特征）
- 对**latent space**进行更进一步的研究

论文总结

A 关键点

- 将CNN网络的最新成果应用到GAN
- 精细的超参数调节尝试
- 对latent space 的多维度分析

B 创新点

- 将GAN应用到表征学习傻姑娘
- 对生成图像中的特特定对象进行擦除
- 对生成图像进行矢量运算

C 启发点

- 一个好的idea，需要靠强大的工程实践来挖掘其潜力
- 调参是一项重要的基础能力
- 表征学习值得关注
- NLP和图像这两大领域，最好都能去有所了解

练习题

- 【思考题】2016年之后，CNN又出现了很多改进，其中还有哪些可以再应用到DCGAN
transformer混合cnn或者纯transformer来替代cnn，加入attention基础
- 【代码实践】对提供的现有代码进行完善，加入模型保存、模型推断代码

```
from torchvision.utils import save_image

epoch = 0 # temporary
batches_done = epoch * len(dataloader) + i
if batches_done % opt.sample_interval == 0:
    save_image(gen_imgs.data[:25], "images/%d.png" % batches_done, nrow=5, normalize=True)
# 保存生成图像

os.makedirs("model", exist_ok=True) # 保存模型
torch.save(generator, 'model/generator.pkl')
torch.save(discriminator, 'model/discriminator.pkl')

print("gen images saved!\n")
print("model saved!")
```

- 【总结】对各种特征正则化方法，以及各种上/下采样方法 分别进行总结

1. L1 & L2范数

首先介绍一下范数的定义，假设 x 是一个向量，它的 L^P 范数定义：

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

在目标函数后面添加一个系数的“惩罚项”是正则化的常用方式，为了防止系数过大从而让模型变得复杂。在加了正则化项之后的目标函数为：

$$\bar{J}(w, b) = J(w, b) + \frac{\lambda}{2m} \Omega(w)$$

式中， $\frac{\lambda}{2m}$ 是一个常数， m 为样本个数， λ 是一个超参数，用于控制正则化程度。

L^1 正则化时，对应惩罚项为 $L1$ 范数：

$$\Omega(w) = \|w\|_1 = \sum_i |w_i|$$

L^2 正则化时，对应惩罚项为 $L2$ 范数：

$$\Omega(w) = \|w\|_2^2 = \sum_i w_i^2$$

从上式可以看出， L^1 正则化通过让原目标函数加上了所有特征系数绝对值的和来实现正则化，而 L^2 正则化通过让原目标函数加上了所有特征系数的平方和来实现正则化。

两者都是通过加上一个和项来限制参数大小，却有不同的效果： L^1 正则化更适用于特征选择，而 L^2 正则化更适用于防止模型过拟合。

2. 数据增强

小幅旋转，平移，放大，缩小，随机选取，噪声等

3. dropout

基本步骤是在每一次的迭代中，随机删除一部分节点，只训练剩下的节点。每次迭代都会随机删除，每次迭代删除的节点也都不一样，相当于每次迭代训练的都是不一样的网络，通过这样的方式降低节点之间的关联性以及模型的复杂度，从而达到正则化的效果。

上采样

别名：放大图像，也叫图像插值。

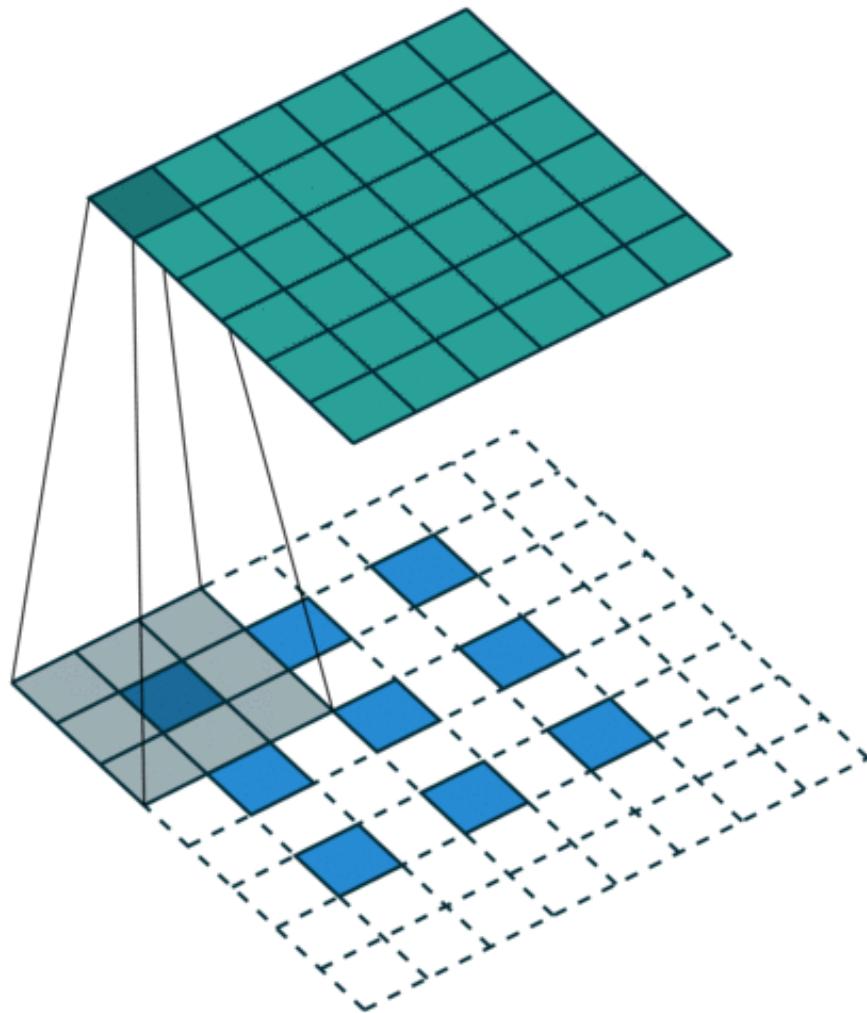
目的：放大原图，从而可以显示在更高分辨率的显示设备上。

缺点：会对图像的质量造成影响，并没有带来更多的信息。

方法：

1. 内插值。插值方法有很多，比如均值，中值，最近邻。通过这种方法，在周围像素色彩的基础上用数学公式计算丢失像素的色彩。
2. 反卷积。即通过转置卷积核的方法来实现卷积的逆过程。
3. 反池化。在池化过程，比如max-pooling时，要记录下每个元素对应kernel中的坐标。反池化时即将每一个元素根据坐标填写，其余位置补0.

下面动图为反卷积操作



双线性插值

双线性插值，又称为双线性内插。在数学上，双线性插值是对线性插值在二维直角网格上的扩展，用于对双变量函数（例如 x 和 y ）进行插值。其核心思想是在两个方向分别进行一次线性插值。

首先在 x 方向上，

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

然后在 y 方向上，

$$\begin{aligned} f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \\ &= \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \right) \end{aligned}$$

...

下采样

别名：缩小图像，降采样

目的：

1. 缩小原图，即生成对应图像的缩略图。
2. 使图像符合对应的显示区域
3. 降低特征的维度并保留有效信息，一定程度上避免过拟合，保持旋转、平移、伸缩不变形。

原理：把一个位于原始图像上的 $s \times s$ 的窗口变成一个像素：

$$p_k = \sum_{i \in \text{win}(k)} I_i / s^2$$

图若为 $x \times y$ ，则下采样之后原图的尺寸为 $(x/s)(y/s)$ 。这说明 s 最好是 x 和 y 的公约数。

实现：池化(pooling)。池化操作是在卷积神经网络中经常采用过的一个基本操作，一般在卷积层后面都会接一个池化操作，使用的比较多的也是 max-pooling 即最大池化，因为 max-pooling 更像是做了特征选择，选出了分类辨识度更好的特征，提供了非线性。

下面动图为卷积操作

