

blog.csdn.net

这些资源你肯定需要！超全的GAN PyTorch+Keras实现集合 - IT派 - CSDN博客

55-69 minutes

选自GitHub

作者：eriklindernoren

机器之心编译

参与：刘晓坤、思源、李泽南

生成对抗网络一直是非常美妙且高效的方法，自 14 年 Ian Goodfellow 等人提出第一个生成对抗网络以来，各种变体和修正版如雨后春笋般出现，它们都有各自的特性和对应的优势。本文介绍了主流的生成对抗网络及其对应的 PyTorch 和 Keras 实现代码，希望对各位读者在 GAN 上的理解与实现有所帮助。

- PyTorch 实现地址：<https://github.com/eriklindernoren/PyTorch-GAN>
- Keras 实现地址：<https://github.com/eriklindernoren/Keras-GAN>

生成对抗网络及其变体的实现分为基于 Keras 和基于 PyTorch 两个版本。它们都是按照原论文实现的，但模型架构并不一定完全和原论文相同，作者关注于实现这些论文最

核心的思想，而并不确定所有层级的配置都和原论文完全一致。本文首先将介绍各种 GAN 的论文摘要，然后提供详细论文和实现的地址。

要使用 PyTorch 或 Keras 实现这些 GAN，我们首先需要下载两个代码仓库，并安装对应所需的依赖包。在终端运行以下命令行将下载 PyTorch-GAN 代码库并安装所需的依赖包：

```
$ git clone https://github.com  
/eriklindernoren/PyTorch-GAN  
$ cd PyTorch-GAN/  
$ sudo pip3 install -r requirements.txt
```

运行以下命令将下载并安装 Keras-GAN 代码库：

```
$ git clone https://github.com  
/eriklindernoren/Keras-GAN  
$ cd Keras-GAN/  
$ sudo pip3 install -r requirements.txt
```

实现

Auxiliary Classifier GAN

- 论文：Conditional Image Synthesis With Auxiliary Classifier GANs
- 作者：Augustus Odena、Christopher Olah 和 Jonathon Shlens
- 论文地址：<https://arxiv.org/abs/1610.09585>
- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN
/blob/master/implementations/acgan/acgan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/acgan/acgan.py)

合成高分辨率的真实感图像一直是机器学习中的长期挑战。

在本论文中，作者介绍了一种用于提升 GAN 在图像合成方面训练效果的新方法。他们构建了一种采用标签条件（label conditioning）的 GAN 变体，这种方法可以生成 128×128 分辨率的图像样本，且能展现出全局一致性。该论文扩展了以前的图像质量评估工作，以提供两个新的分析来评估类别条件（class-conditional）图像合成模型中样本的辨识度和多样性。在 ImageNet 的 1000 个类别中， 128×128 的样本要比手动调整为 32×32 的样本高两倍多的可辨识度。此外，84.7% 的类别具有与 ImageNet 真实图像相媲美的样本。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练 acgan：

```
$ cd implementations/acgan/  
$ python3 acgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd acgan/  
$ python3 acgan.py
```

对抗自编码器（Adversarial Autoencoder）

- 论文：Adversarial Autoencoders
- 作者：Alireza Makhzani、Jonathon Shlens、Navdeep Jaitly、Ian Goodfellow 和 Brendan Frey
- 论文地址：<https://arxiv.org/abs/1511.05644>
- 代码地址：https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/aae/adversarial_autoencoder.py

在这篇论文中，作者们提出了「对抗自编码器」（AAE），它是一种概率自编码器，且通过最近提出的生成对抗网络实现变分推断。简单而言，即通过匹配自编码器隐藏编码向量的聚合后验分布（aggregated posterior）和任意的先验分布。匹配聚合后验分布和先验分布确保了从任意先验空间中的采样都能生成有意义的样本。最后，对抗自编码器的解码器将学习一个深度生成模型以将先验分布映射到数据分布中。作者们展示了对抗自编码器如何应用于半监督分类、图像内容和风格解析、无监督聚类、降维算法和数据可视化等内容。作者还实现了一些实验，并在生成建模和半监督分类任务中取得了很好的性能。

如果当前地址为 PyTorch-GAN/，那么你可以输入以下命令行，用 PyTorch 开始训练 AAE：

```
$ cd implementations/aae/  
$ python3 aae.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd aae/  
$ python3 aae.py
```

Boundary-Seeking GAN

- 论文：Boundary-Seeking Generative Adversarial Networks
- 作者：R Devon Hjelm、Athul Paul Jacob、Tong Che、Adam Trischler、Kyunghyun Cho 和 Yoshua Bengio
- 论文地址：<https://arxiv.org/abs/1702.08431>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/bgan/bgan.py>

生成对抗网络（GAN）是一种学习框架，它依赖训练判别器来度量目标和生成分布之间的差距。正如通常所说的，GAN 依赖于生成的样本对生成器的参数是完全可微的，因此它是适用于离散数据。作者引入了一种用离散数据训练 GAN 的方法，该方式使用判别器的差异度量来计算生成样本的重要权重，从而为训练生成器提供策略梯度。此外，重要权重与判别器的决策边界有非常强的联系，因此作者们称这种方法为 boundary-seeking GAN (BGAN)。他们证明了该算法在离散图像和字符级的自然语言生成任务上具有高效性。此外，搜索判别器边界的目标可扩展到连续数据，并用来提升训练的稳定性。最后，该论文还展示了在 Celeba、大规模卧室场景理解 (LSUN) 和不带条件的 ImageNet 上具有优秀的性能。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将开始训练 BGAN：

```
$ cd implementations/bgan/  
$ python3 bgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd bgan/  
$ python3 bgan.py
```

Context-Conditional GAN

- 论文：Boundary-Seeking Generative Adversarial Networks
- 作者：Emily Denton、Sam Gross 和 Rob Fergus
- 论文地址：<https://arxiv.org/abs/1702.08431>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN>

/blob/master/implementations/ccgan/ccgan.py

本论文介绍了一种基于图像的半监督学习方法，它在绘画中将使用对抗性损失函数。我们将删除随机图像块后的图像馈送给生成器，而生成器的任务就是根据周围的像素填补这些删除的图像块。然后将绘制后的图像展现给判别器网络，并判断它们是否是真实图像。这一任务可充当判别器标准监督训练的正则化器。使用这种方法，我们能以半监督的方式直接训练大规模 VGG 风格的网络。最后，作者们还在 STL-10 和 PASCAL 数据集上评估了该方法，它的结果能至少获得当前业内最优的水平。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将开始训练 CCGAN：

```
$ cd implementations/ccgan/  
$ python3 ccgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd ccgan/  
$ python3 ccgan.py
```

条件 GAN

- 论文：Conditional Generative Adversarial Nets
- 作者：Mehdi Mirza 和 Simon Osindero
- 论文地址：<https://arxiv.org/abs/1411.1784>
- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN
/blob/master/implementations/cgan/cgan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/cgan/cgan.py)

生成对抗网络近来被引入并作为训练生成模型的新方法。在本论文的研究工作中，作者们介绍了生成对抗网络的条件

约束版，即可以通过数据 y 来构建生成网络，而我们期望生成器和判别器能加上约束条件。作者表示模型可以以类别标签为条件生成 MNIST 手写数字，同时还展示了如何使用该模型学习多模态模型，并提供了一个应用于图像标注的简单示例，他们展示了这种方法如何生成不属于训练标注的描述性标签。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将开始训练 CGAN：

```
$ cd implementations/cgan/  
$ python3 cgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

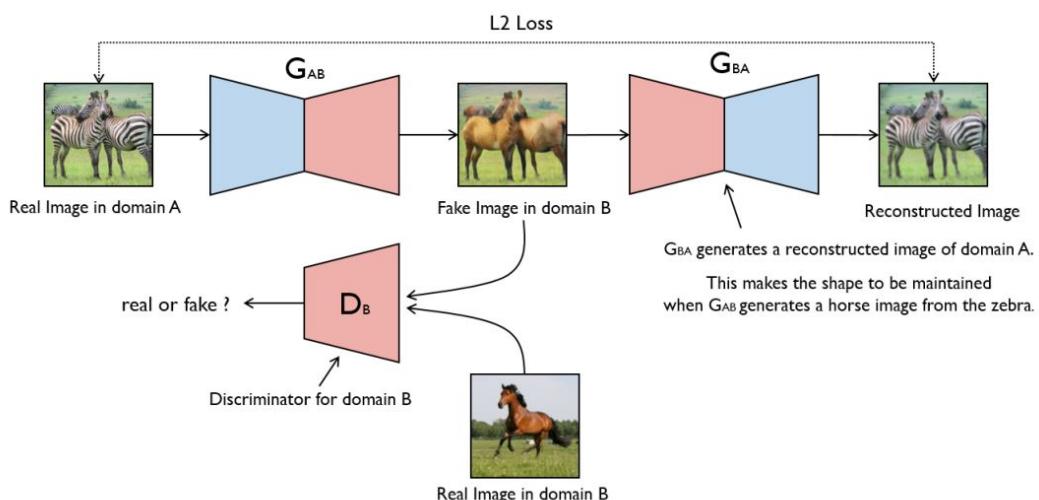
```
$ cd cgan/  
$ python3 cgan.py
```

CycleGAN

- 论文：Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- 作者：Jun-Yan Zhu、Taesung Park、Phillip Isola 和 Alexei A. Efros
- 论文地址：<https://arxiv.org/abs/1703.10593>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/cyclegan/cyclegan.py>

图像到图像变换是一种视觉和图像问题，其目标是使用分配图像配对的训练集学习输入图像和输出图像之间的映射。然而，对于很多任务来说，配对的训练数据是不可获取的。作者提出了一种新方法，在没有配对示例的限制下，学习从源

域 X 到目标域 Y 的图像变换。其目标是学习一个映射： $X \rightarrow Y$ ，通过对抗损失使得 $G(X)$ 中的图像分布和 Y 的分布是不可区分的。由于这个映射是高度受限的，作者将其和一个逆映射 $F: Y \rightarrow X$ 耦合，并引入了一个周期一致损失来迫使 $F(G(X)) \approx X$ （反之亦然）。研究者在无配对训练数据的多项任务（包括款式风格迁移、目标变形、季节迁移、照片增强等）上做了定性实验。并且和多种之前方法的定量结果表明，该方法在性能上有优越性。



如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd data/
$ bash download_cyclegan_dataset.sh
apple2orange
$ cd ../implementations/cyclegan/
$ python3 cyclegan.py --dataset_name
apple2orange
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd cyclegan/
```

```
$ bash download_dataset.sh apple2orange  
$ python3 cyclegan.py
```



第一行展示了两个域的原始图像。第二行展示了两个图像的变换版本。第三行展示了重构图像。

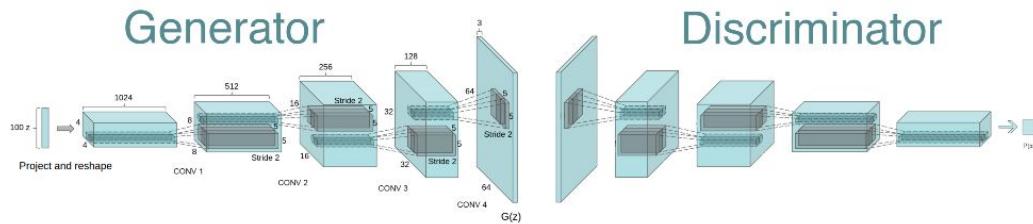
Deep Convolutional GAN

- 论文：Deep Convolutional Generative Adversarial Network
- 作者：Alec Radford、Luke Metz 和 Soumith Chintala
- 论文地址：<https://arxiv.org/abs/1511.06434>
- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN /blob/master/ implementations/dcgan/dcgan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/ implementations/dcgan/dcgan.py)

在 GAN 的第一篇论文出来之后的大概一年时间里，训练 GAN 与其说是科学，倒不如说是艺术——模型很不稳定，需要大量调整才能工作。2015 年时，Radford 等人发表了题为《使用深度卷积生成对抗网络的无监督表征学习

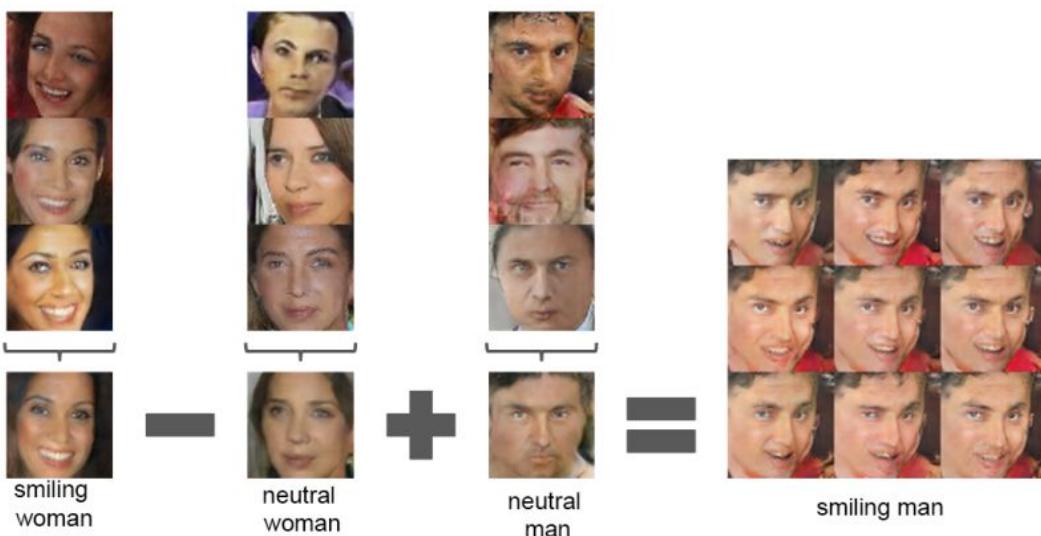
(Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks) » 的论文，描述了之后被称为 DCGAN 的著名模型。

在 DCGAN 这项工作中，作者们引入一种带有结构约束的卷积神经网络，并证明深度卷积对抗网络从目标到场景能学习层级表征。



图片来自：Radford et al., 2015

关于 DCGAN，最值得一提的是这个架构在大多数情况下都是稳定的。这是第一篇使用向量运算描述生成器学习到的表征的固有性质的论文：这与 Word2Vec 中的词向量使用的技巧一样，但却是对图像操作的！



图片来自：Radford et al., 2015

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/dcgan/
$ python3 dcgan.py
```

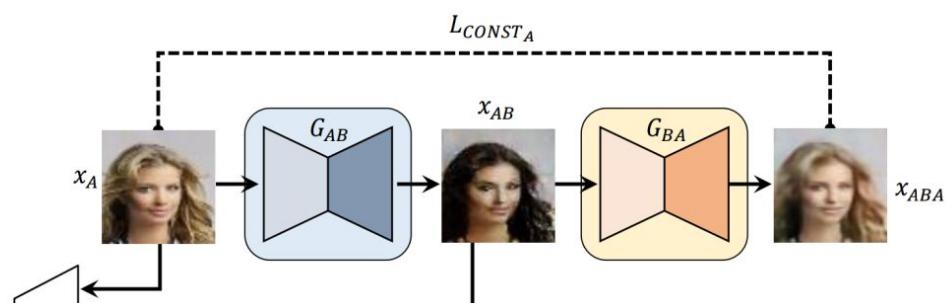
如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

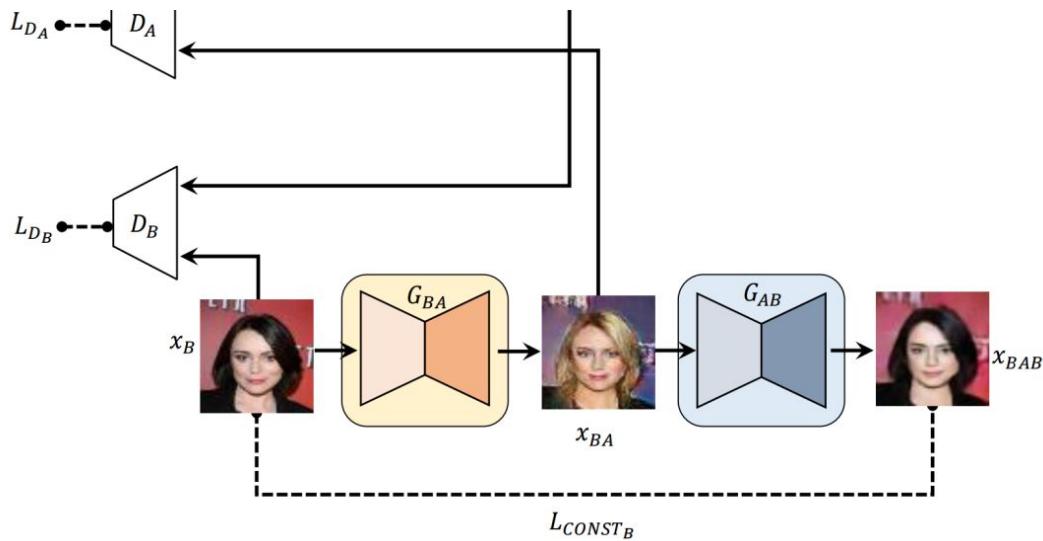
```
$ cd dcgan/
$ python3 dcgan.py
```

DiscoGAN (学习用生成对抗网络发现跨域关系)

- 论文：Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- 作者：Taeksoo Kim、Moonsu Cha、Hyunsoo Kim、Jung Kwon Lee 和 Jiwon Kim
- 论文地址：<https://arxiv.org/abs/1703.05192>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/discogan/discogan.py>

人类可以轻易地识别不同领域的数据之间的关系而不需要任何的监督，但让机器学习自动化地发现这些关系是很大的挑战并需要很多真实的配对来展示这些关系。为了避免代价昂贵的配对工程，通过给定的非配对数据来发现跨域关系，作者提出了基于生成对抗网络的方法来学习发现不同领域之间的关系，即 DiscoGAN。使用发现的关系，该网络可以成功地将一个领域的风格迁移到另一个上，同时保持关键的属性，例如定向和面部身份。





如果当前地址为 PyTorch-GAN/, 那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd data/
$ bash download_pix2pix_dataset.sh
edges2shoes
$ cd ../implementations/discogan/
$ python3 discogan.py --dataset_name
edges2shoes
```

如果当前地址为 Keras-GAN/, 那么我们需要使用 Keras 实现训练：

```
$ cd discogan/
$ bash download_dataset.sh edges2shoes
$ python3 discogan.py
```

DRAGAN (改善 GAN 的收敛性和稳定性)

- 论文：On Convergence and Stability of GANs
- 作者：Naveen Kodali、Jacob Abernethy、James Hays 和 Zsolt Kira
- 论文地址：<https://arxiv.org/abs/1705.07215>

- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN
/blob/master/implementations/dragan/dragan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/dragan/dragan.py)

通过将 GAN 的训练动态看成遗憾最小化过程，这和流行的角度相反（真实和生成分布之间的偏离存在一致最小化）。作者从新的视角分析了 GAN 训练过程的收敛行为，易理解模式崩溃的原因。他们假定在非凸优化过程中出现的局域均衡是导致模式崩溃的原因。研究表明这些局域均衡通常导致判别器函数在某些真实数据点处的尖锐梯度，而使用一种称为 DRAGAN 的梯度惩罚方案可以避免这些退化的局域均衡。DRAGAN 可以让训练的速度更快，模型获得更高的稳定性，以及更少的模式崩溃，在多种架构和目标函数的生成器网络上得到更优的建模性能。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/dragan/  
$ python3 dragan.py
```

DualGAN

- 论文：DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- 作者：Zili Yi、Hao Zhang、Ping Tan 和 Minglun Gong
- 论文地址：<https://arxiv.org/abs/1704.02510>
- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN
/blob/master/implementations/dualgan/dualgan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/dualgan/dualgan.py)

跨域图像到图像变换的条件生成网络在近期取得了很大的进步。由于任务的复杂度，训练一个条件 GAN 需要成千上百万张标记的图像。然而，人类标记是代价昂贵的，甚至是不

可行的，并且很多数据可能是不可用的。受自然图像变换的对偶学习启发，一种新型的 Dual-GAN 机制被提出，它允许在两个域的无标记图像集训练以实现图像变换。在该架构中，原始 GAN 学习从域 U 向域 V 的图像变换，同时，对偶的 GAN 学习将任务倒转。由原始任务和对偶任务构成的循环允许图像从任意两个域之一被变换然后被重构。因此可以用关于重构误差的损失函数来训练变换器。在多个使用无标签数据的图像变换任务上的实验表明，DualGAN 相比单个 GAN 可以取得相当好的结果。对于某些任务，DualGAN 甚至可以得到相当或稍微超越条件 GAN 在全标记数据上的结果。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd data/  
$ bash download_pix2pix_dataset.sh facades  
$ cd ../implementations/dualgan/  
$ python3 dualgan.py --dataset_name facades
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd dualgan/  
$ python3 dualgan.py
```

GAN

- 论文：Generative Adversarial Network
- 作者：Ian J. Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville 和 Yoshua Bengio
- 论文地址：<https://arxiv.org/abs/1406.2661>

- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN
/blob/master/implementations/gan/gan.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/gan/gan.py)

GAN 最初由 Ian Goodfellow 提出，它有两个网络：生成器和鉴别器。两个网络在同一时间进行训练，并在极小极大（minimax）游戏中相互博弈。生成器通过创建逼真的图像来试图欺骗鉴别器，而鉴别器被训练从而不被生成器所愚弄。首先，生成器生成图像。它通过从简单分布中（例如正态分布）采样向量噪声 Z ，然后将该矢量上采样到图像来生成图像。在第一次迭代中，这些图像看起来很嘈杂。然后，鉴别器被给予真、假图像，并学习区分它们。生成器稍后通过反向传播步骤接收鉴别器的「反馈」，在产生图像时变得更好。最后，我们希望假图像的分布尽可能接近真实图像的分布。或者，简单来说，我们希望假图像看起来尽可能貌似真实。

值得一提的是，由于 GAN 中使用的极小极大（minimax）优化，训练有可能相当不稳定。但是，有一些技巧可以用来使得训练更鲁棒。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/gan/  
$ python3 gan.py
```

Keras 示例 1：如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd gan/  
$ python3 gan.py
```

Keras 示例 2：如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd gan/  
$ python3 gan_rgb.py
```

Least Squares GAN

- 论文：Least Squares Generative Adversarial Networks
- 作者：Xudong Mao、Qing Li、Haoran Xie、Raymond Y.K. Lau、Zhen Wang 和 Stephen Paul Smolley
- 论文地址：<https://arxiv.org/abs/1611.04076>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/lsgan/lsgan.py>

最近应用生成对抗网络 (generative adversarial networks/GAN) 的无监督学习被证明是十分成功且有效的。常规生成对抗网络假定作为分类器的辨别器是使用 sigmoid 交叉熵损失函数 (sigmoid cross entropy loss function)。然而这种损失函数可能在学习过程中导致梯度消失 (vanishing gradient) 问题。为了克服这一困难，我们提出了最小二乘生成对抗网络 (Least Squares Generative Adversarial Networks/LSGANs)，该生成对抗网络的辨别器 (discriminator) 采用最小平方损失函数 (least squares loss function)。我们也表明 LSGAN 的最小化目标函数 (objective function) 服从最小化 Pearson X^2 divergence。LSGAN 比常规生成对抗网络有两个好处。首先 LSGAN 能够比常规生成对抗网络生成更加高质量的图片。其次 LSGAN 在学习过程中更加地稳定。我们在五个事件数据集 (scene datasets) 和实验结果上进行评估，结果证明由 LSGAN 生成的图像看起来比由常规 GAN 生成的图像更加真实一些。我们还对 LSGAN 和常规 GAN 进行了两个比较实验，其证明了 LSGAN 的稳定性。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/lsgan/  
$ python3 lsgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

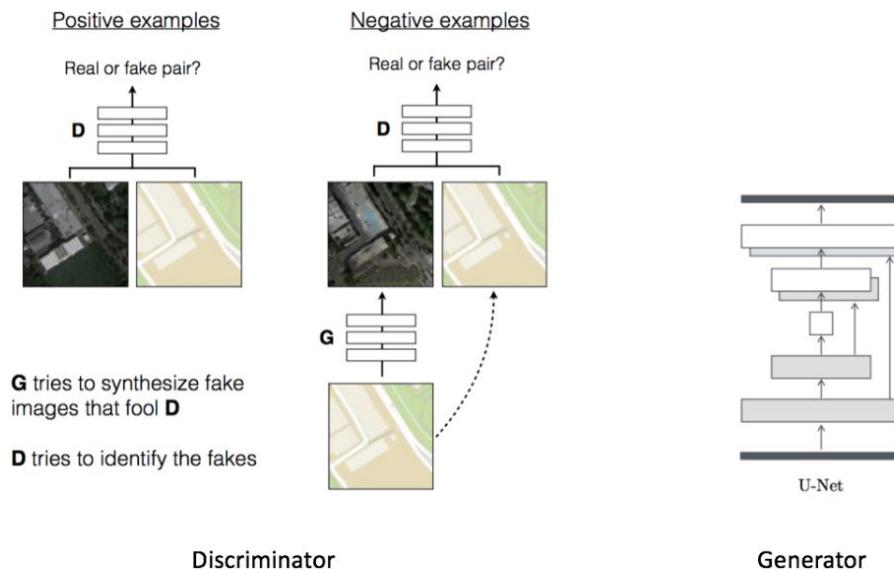
```
$ cd lsgan/  
$ python3 lsgan.py
```

Pix2Pix

- 论文：Unpaired Image-to-Image Translation with Conditional Adversarial Networks
- 作者：Phillip Isola、Jun-Yan Zhu、Tinghui Zhou 和 Alexei A. Efros
- 论文地址：<https://arxiv.org/abs/1611.07004>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/pix2pix/pix2pix.py>

Pix2Pix 探索了将条件 GAN 作为图像到图像变换问题的通用解决方案的可行性。这些网络不仅能学习从输入图像到输出图像的变换，还能学习一个损失函数来训练这个损失。这使其可以应用相同的一般性方法到传统上需要非常不同的损失函数的问题上。研究表明该方法在从标签映射合成照片、从边缘映射重构图像，以及图像上色等任务上非常有效。实际上，由于和 Pix2Pix 论文相关的 pix2pix 软件的发布，已经有大量的网络用户（其中包括很多艺术家）发布了用该系统处理的实验结果，进一步展示了它的广泛应用价值和不需要参数调整的易用性。不需要手动修改映射函数和损失函数，该

方法就可以取得很优越的结果。

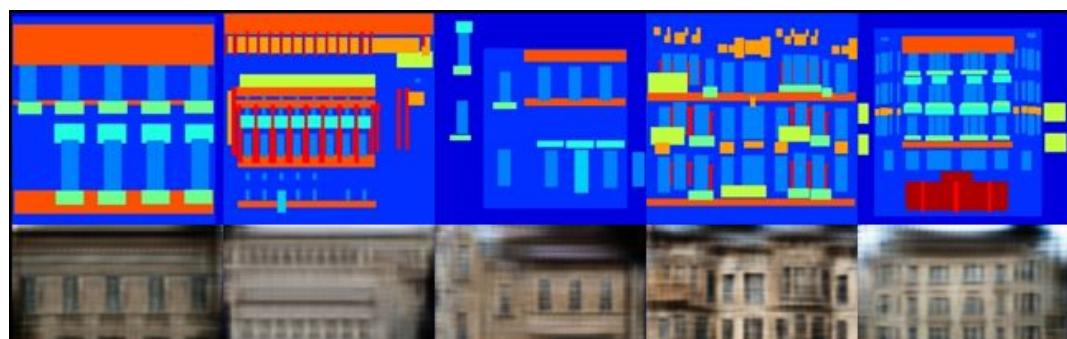


如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd data/
$ bash download_pix2pix_dataset.sh facades
$ cd ../implementations/pix2pix/
$ python3 pix2pix.py --dataset_name facades
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd pix2pix/
$ bash download_dataset.sh facades
$ python3 pix2pix.py
```





第一行展示了生成器的条件，第二行展示了输出，第三行展示了条件对应的真实图像。

PixelDA

- 论文：Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks
- 作者：Konstantinos Bousmalis、Nathan Silberman、David Dohan、Dumitru Erhan 和 Dilip Krishnan
- 论文地址：<https://arxiv.org/abs/1612.05424>
- 代码地址：[https://github.com/eriklindernoren/PyTorch-GAN /blob/master/implementations/pixelda/pixelda.py](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/pixelda/pixelda.py)

对于很多任务而言，收集标注良好的图像数据集来训练机器学习算法是代价昂贵的。一个有吸引力的替代方案是渲染合成数据，其中真实标签是自动生成的。不幸的是，纯粹在渲染图像上训练的模型通常无法泛化到真实图像上，为了解决这个弊端，以前的研究通过引入无监督域适应算法来实现两个领域的表征映射，并学习提取域不变的特征。在 PixelDA 中，作者提出了一个新的方法，以无监督的方式在像素空间中实现域变换。该基于生成对抗网络的方法将源域的图像渲染成像是来自目标域的图像。该方法不仅能生成可信的样本，还在多个无监督域适应场景中大幅超越了当前最佳方法。最后，研究表明该适应过程可以泛化到训练过程中未见过的目标类别。

从 MNIST 到 MNIST-M 的分类性能

PixelDA 在 MNIST 上训练一个分类器，并可以变换到和 MNIST-M 相似的图像（通过执行无监督的图像到图像域适应）。该模型和在 MNIST 上训练分类器并在 MNIST-M 上评估的朴素方法对比。朴素方法在 MNIST 上取得了 55% 的准确率，而在域适应上训练的模型获得了 95% 的分类准确率。

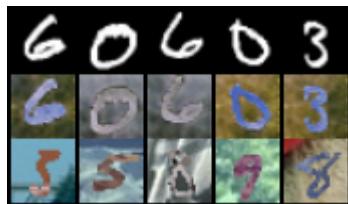
如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/pixelda/  
$ python3 pixelda.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd pixelda/  
$ python3 pixelda.py
```

Method	Accuracy
Naive	55%
PixelDA	95%



第一行展示了 MNIST 域的原始图像。中间行展示了第一行图像的变换版本。最后一行展示了 MNIST-M 域的图像示例。

Semi-Supervised GAN

- 论文：Semi-Supervised Learning with Generative Adversarial Networks
- 作者：Augustus Odena
- 论文地址：<https://arxiv.org/abs/1606.01583>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/srgan/srgan.py>

半监督生成对抗网络通过强制判别器网络输出类别标签将生成对抗网络（GAN）扩展到半监督情境中。论文作者在一个数据集上训练了一个生成器 G 和判别器 D，其中每个输入属于 N 个类别之一。在训练时，D 被用于预测输入属于 N+1 个类别的哪一个，其中额外的类别对应于 G 的输出。研究表明该方法可以用于构建一个数据高效的分类器，相比于常规的 GAN，它可以生成更加高质量的样本。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/srgan/  
$ python3 srgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd srgan/  
$ python3 srgan.py
```

Super-Resolution GAN

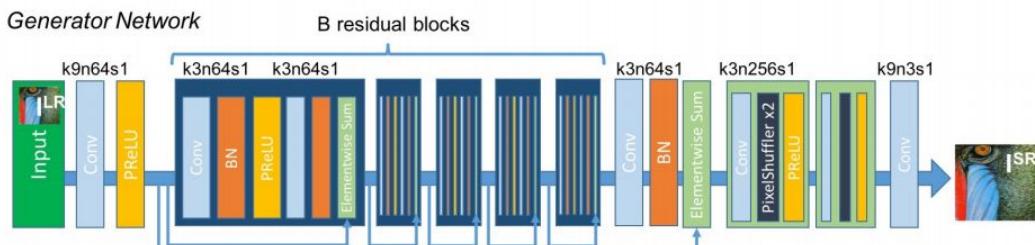
- 论文：Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
- 作者：Christian Ledig、Lucas Theis、Ferenc Huszar、Jose Caballero、Andrew Cunningham、Alejandro Acosta、

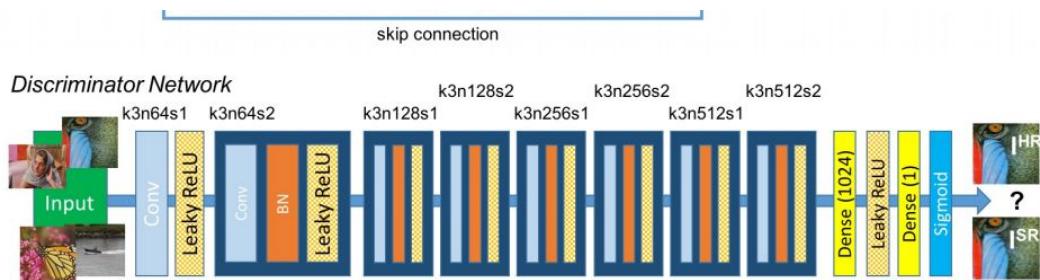
Andrew Aitken、Alykhan Tejani、Johannes Totz、Zehan Wang 和 Wenzhe Shi

- 论文地址：<https://arxiv.org/abs/1609.04802>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/srgan/srgan.py>

尽管使用更快和更深的卷积神经网络在单张图像超分辨率上已经得到了准确率和速度的突破，仍然有一个中心问题为得到解决：如何在大的粗化（upsampling）系数时的超分辨率恢复精细纹理细节？基于优化的超分辨率方法的行为在原则上是由目标函数的选择驱动的。近期的研究主要聚焦于最小化重构均方误差。得到的评估结果具有很高峰值的信噪比，但它们通常缺少高频细节，并无法在高分辨率上匹配期望的保真度。

SRGAN 是一个用于图像超分辨率的生成对抗网络。这是首个能在 4x 粗化系数上推断照片级自然图像的框架。为了达到这个目的，论文作者提出了一个感知损失函数，它一个对抗损失和一个内容损失构成。通过训练判别网络来区分超分辨图像和原始照片级图像，对抗损失迫使网络生成自然图像流形。此外，内容损失是通过感知相似性驱动的而不是像素空间的相似性。该架构使用的深度残差网络可以从公开基准的严重下采样的图像上恢复照片级纹理。扩展的平均意见分数（MOS）测试表明，使用 SRGAN 可以大大提高感知质量。使用 SRGAN 获得的 MOS 分数和那些原始高分辨率图像很相近，高于所有当前最佳方法。





如果当前地址为 PyTorch-GAN/, 那么使用以下命令行将使用 PyTorch 开始训练:

```
$ cd implementations/srgan/
```

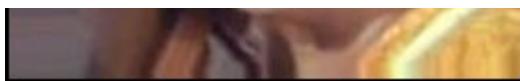
```
$ python3 srgan.py
```

如果当前地址为 Keras-GAN/, 那么我们需要使用 Keras 实现训练:

```
$ cd srgan/
```

```
$ python3 srgan.py
```





第一行由 SRGAN 生成，第二行是全分辨率图像。

Wasserstein GAN (WGAN)

- 论文：Wasserstein GAN
- 作者：Martin Arjovsky, Soumith Chintala, Léon Bottou
- 论文地址：<https://arxiv.org/abs/1701.07875>
- 代码地址：<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/wgan/wgan.py>

通过改变损失函数以包含 Wasserstein 距离，WGAN 具有与图像质量相关的损失函数。此外，训练稳定性也提高了，而且不依赖于架构。

GAN 一直在收敛性方面存在问题，结果是，你不知道何时停止训练。换句话说，损失函数与图像质量不相关。这是一个头痛的大问题，因为：

- 你需要不断查看样本，以了解你的模型是否在正确训练。
- 你不知道何时应该停止训练（没有收敛）。
- 你没有一个量化数值告诉你调整参数的效果如何。

GAN 可被解释为最小化 Jensen-Shannon 发散，如果真和假的分布不重叠（通常是这种情况），则它为 0。所以，作者使用了 Wasserstein 距离，而不是最小化 JS 发散，它描述了从一个分布到另一个分布的「点」之间的距离。因此，WGAN 具有与图像质量相关的损失函数并能够实现收敛。它也更加稳定，也就意味着它不依赖于架构。例如，即使你去掉批处理归一化或尝试奇怪的架构，它也能很好地工作。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/wgan/  
$ python3 wgan.py
```

如果当前地址为 Keras-GAN/，那么我们需要使用 Keras 实现训练：

```
$ cd wgan/  
$ python3 wgan.py
```

Wasserstein GAN GP

- 论文：Improved Training of Wasserstein GANs
- 作者：Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville
- 论文地址：<https://arxiv.org/abs/1704.00028>
- 代码地址：https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/wgan_gp/wgan_gp.py

生成对抗网络（GAN）是很强大的生成模型，但会遭遇训练不稳定的问题。最近提出的 WGAN 提高了 GAN 的训练稳定性，但有时候仍然会生成低质量的样本或无法收敛。论文作者发现这些问题是由 WGAN 中使用的权重修剪，以强制在判别器上添加一个 Lipschitz 约束，这会导致不希望出现的行为。他们提出了权重修剪的替代方案：惩罚判别器的关于它的输入的梯度范数。该方法相比标准的 WGAN 表现更好，在多种 GAN 架构中实现稳定的训练，而几乎不需要超参数的调整，包括 101 层的 ResNet 和离散数据上的语言模型。该方法可以在 CIFAR-10 和 LSUNbedrooms 数据集上生成高质量的图像。

如果当前地址为 PyTorch-GAN/，那么使用以下命令行将使用 PyTorch 开始训练：

```
$ cd implementations/wgan_gp/  
$ python3 wgan_gp.py
```

oooooooooooo



IT派 - {技术青年圈}持续关注互联网、区块链、人工智能领域



公众号回复“机器学习”，

邀你加入{ IT派AI机器学习群 }