人工智能之深度学习

生成对抗网络GAN

主讲人：刘老师(GerryLiu)

课程要求

- 课上课下“九字”真言
  - 认真听，**善摘录，勤思考**
  - **多温故，乐实践**，再发散
- 四不原则
  - 不懒散惰性，不迟到早退
  - 不请假旷课，不拖延作业
- 一点注意事项
  - 违反“四不原则”，不推荐就业

课程内容

- 什么是生成式对抗网络
- 生成式对抗网络(GAN)
  - 应用场景
  - 层次结构
  - GAN描述
- 实现与训练
- 案例分析

什么是生成式对抗网络

- 生成式对抗网络（GAN）是一种深度学习模型，是近年来复杂分布上无监督学习最具前景的方法之一。模型通过框架中（至少）两个模块：生成模型（Generative Model）和判别模型（Discriminative Model）的互相博弈学习产生相当好的输出。原始 GAN 理论中，并不要求 G 和 D 都是神经网络，只需要是能拟合相应生成和判别的函数即可。但实用中一般均使用深度神经网络作为 G 和 D 。捕获数据分布的生成模型G，和估计样本来自训练数据的概率的判别模型D。G的训练程序是将D错误的概率最大化。

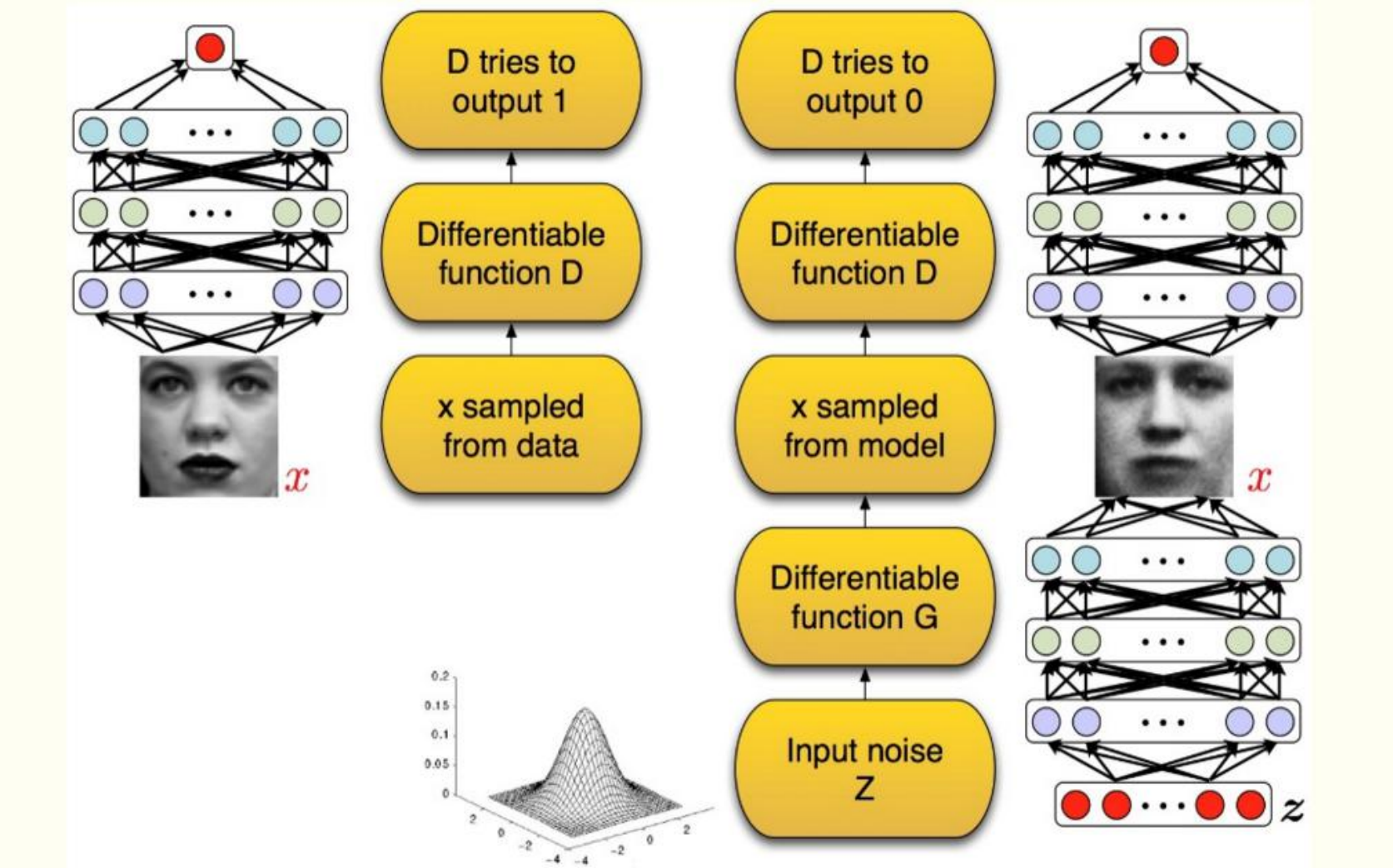
什么是生成式对抗网络

- 机器学习的模型可大体分为两类，生成模型（Generative Model）和判别模型（Discriminative Model）。判别模型需要输入变量，通过某种模型来预测。生成模型是给定某种隐含信息，基于学习得到的数据分布信息来随机产生观测数据。如：
  - 判别模型：给定一张图，判断这张图里的动物是猫还是狗
  - 生成模型：给一系列猫的图片，生成一张新的猫咪（不在数据集里）
- 生成式对抗网络将机器学习中的两大类模型，**G**enerative和**D**iscrimitive给紧密地联合在了一起

什么是生成式对抗网络

- 假设我们有两个网络，G（Generator）和D（Discriminator）。它们的功能分别是：
- G是一个生成图片的网络，它接收一个随机的噪声z，通过这个噪声生成图片，记做G(z)。
- D是一个判别网络，判别一张图片是不是“真实的”。它的输入参数是x，x代表一张图片，输出D（x）代表x为真实图片的概率，如果为1，就代表100%是真实的图片，而输出为0，就代表不可能是真实的图片。
- 在训练过程中，生成网络G的目标就是尽量生成真实的图片去欺骗判别网络D。而D的目标就是尽量把G生成的图片和真实的图片分别开来。这样，G和D构成了一个动态的“博弈过程”。
- 最后博弈的结果是什么？在最理想的状态下，G可以生成足以“以假乱真”的图片G(z)。对于D来说，它难以判定G生成的图片究竟是不是真实的，因此D(G(z)) = 0.5。

什么是生成式对抗网络


D tries to output 1
Differentiable function D
x sampled from data
D tries to output 0
Differentiable function D
x sampled from model
Differentiable function G
Input noise Z
x
x
z

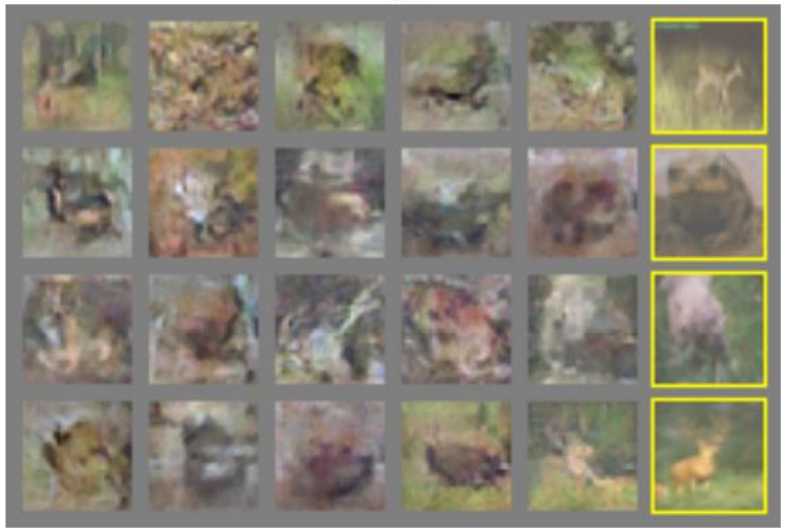
什么是生成式对抗网络

- 最右边的一列是真实样本的图像，前面五列是生成网络生成的样本图像，可以看到生成的样本还是很像真实样本的，只是和真实样本属于不同的类，类别是随机的。

生成式对抗网络的应用

a)

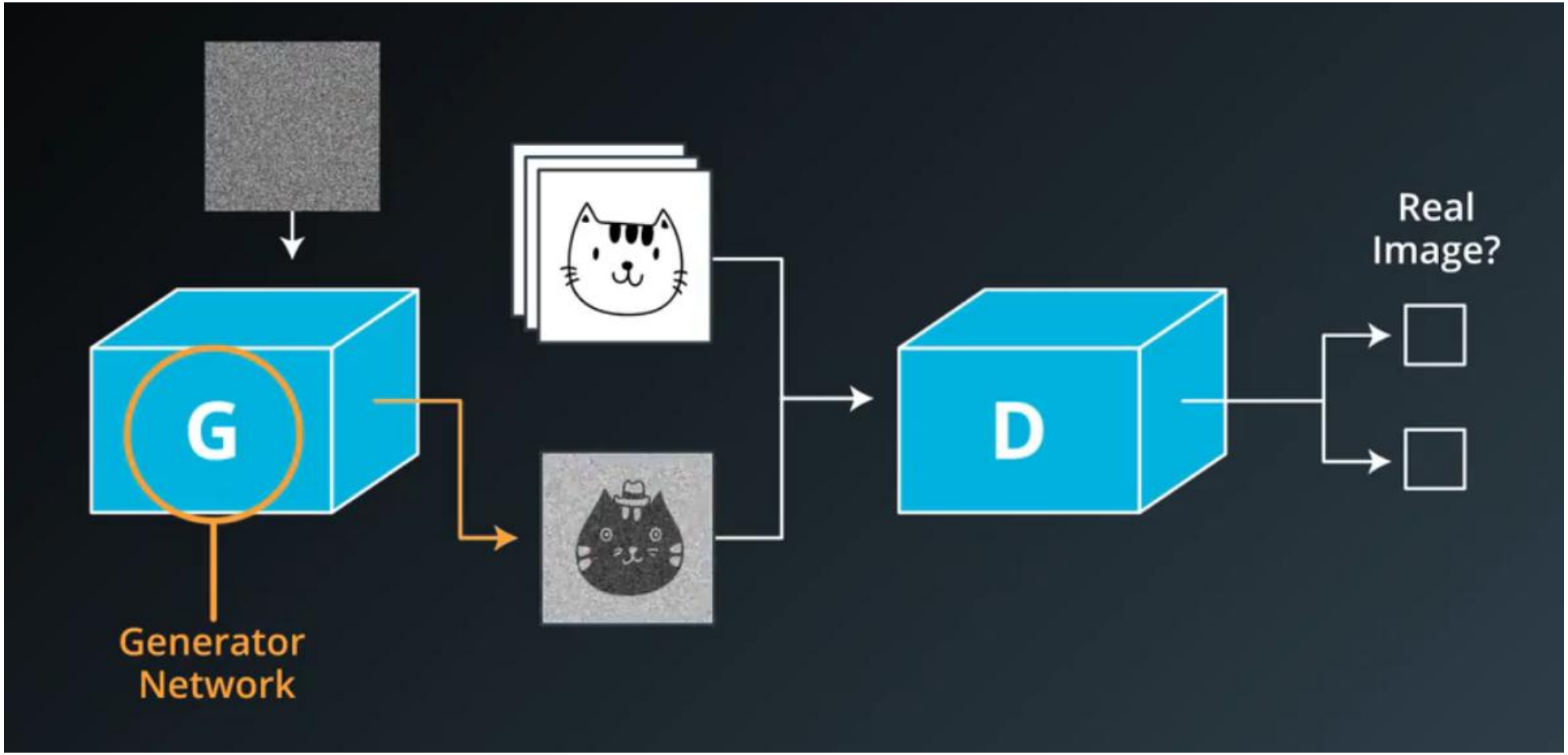
b)



生成式对抗网络(GAN)应用场景

- 图像生成，超分辨率
- 语义分割
- 文字生成
- 数据增强
- 聊天机器人
- 信息检索，排序

神经网络之结构



生成式对抗网络

- GAN是什么，他是怎么对抗的?
  - 生成式对抗网络包含一个**生成模型**（generative model，G）和一个**判别模型**（discriminative model，D）。
  - 主要解决的问题就是如何从训练样本中学习出新样本。
  - 生成模型就是负责训练出样本的分布，判别模型是一个二分类器，用来判断输入样本时真实数据还是训练生成的样本。

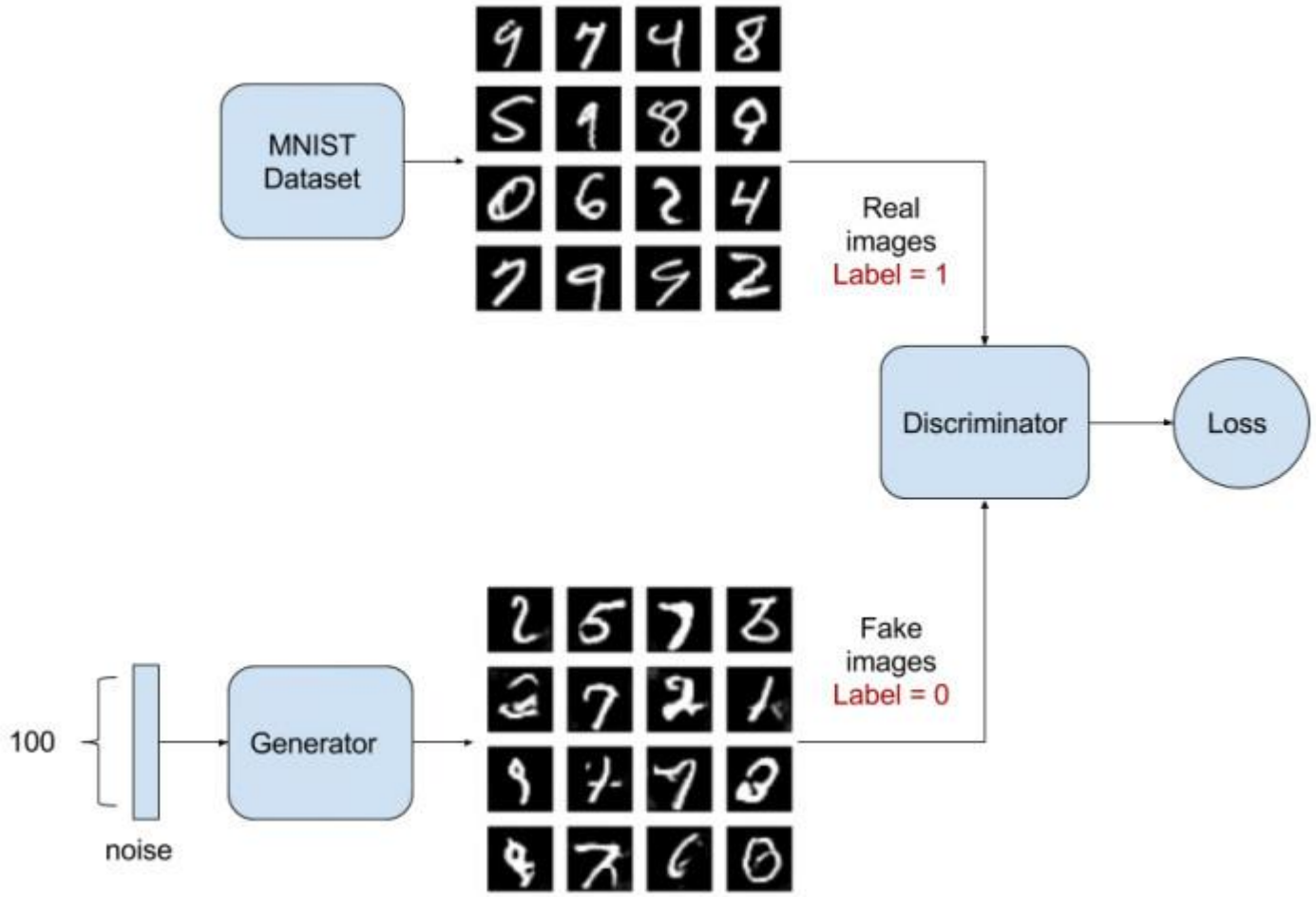
优势

- **GAN是更好的生成模型，在某种意义上避免了马尔科夫链式的学习机制**，这使得它能够区别于传统的概率生成模型。传统概率生成模型一般都需要进行马可夫链式的采样和推断，而GAN避免了这个计算复杂度特别高的过程，直接进行采样和推断，从而提高了GAN的应用效率，所以其实际应用场景也就更为广泛。
- 其次**GAN是一个非常灵活的设计框架**，各种类型的损失函数都可以整合到GAN模型当中，这样使得针对不同的任务，我们可以设计不同类型的损失函数，都会在GAN的框架下进行学习和优化。
- 再次，最重要的一点是，当概率密度不可计算的时候，传统依赖于数据自然性解释的一些生成模型就不可以在上面进行学习和应用。**但是GAN在这种情况下依然可以使用，这是因为GAN引入了一个非常聪明的内部对抗的训练机制**，可以逼近一些不是很容易计算的目标函数。

基本框架

- 一个最朴素的GAN模型，实际上是将一个随机变量（可以是高斯分布，或0到1之间的均匀分布），通过参数化的概率生成模型（通常是用一个**神经网络**模型来进行参数化），进行概率分布的逆变换采样，从而得到一个生成的概率分布。
- GAN的或者一般概率生成模型的训练目的，就是要使得生成的概率分布和真实数据的分布尽量接近，从而能够解释真实的数据。但是在实际应用中，我们完全没有办法知道真实数据的分布。我们所能够得到的只是从这个真实的数据分布中所采样得到的一些真实数据。

生成式对抗神经网络GAN-结构



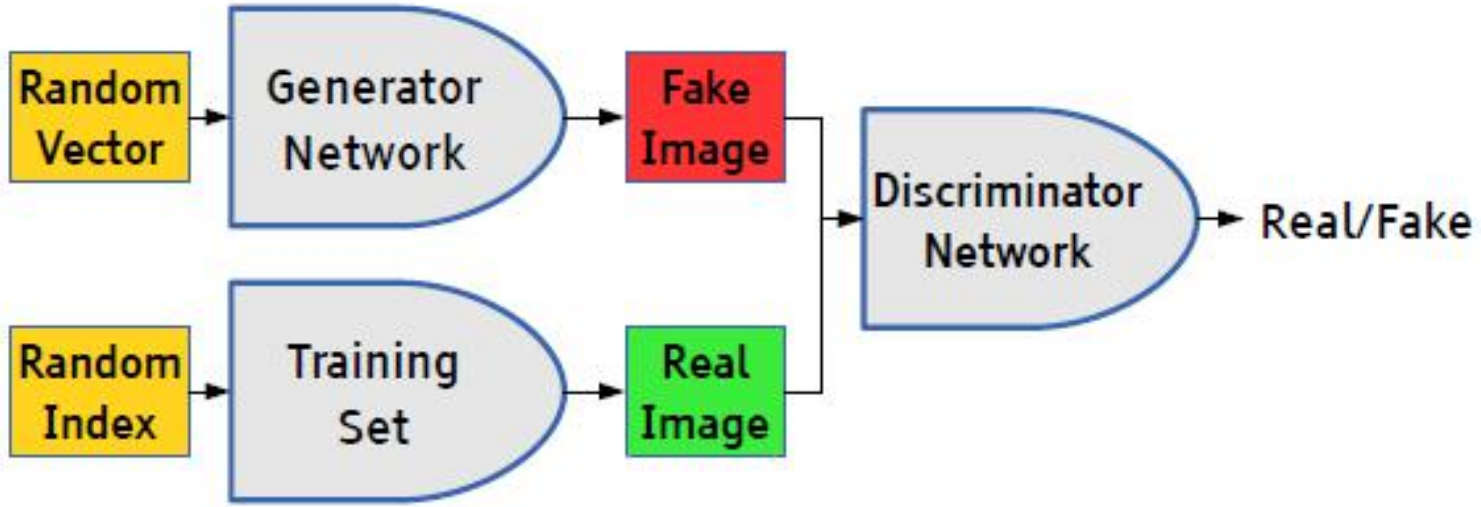
怎么定义损失

- 传统的生成模型，一般都采用数据的似然性来作为优化的目标，但**GAN创新性地使用了另外一种优化目标。**
- 首先，它引入了一个判别模型（常用的有支持向量机和多层神经网络）。
- 其次，它的优化过程就是在寻找生成模型和判别模型之间的一个纳什均衡。
- GAN所建立的一个学习框架，**实际上就是生成模型和判别模型之间的一个模仿游戏。**生成模型的目的，就是要尽量去模仿、建模和学习真实数据的分布规律；而判别模型则是要判别自己所得到的一个输入数据，究竟是来自于真实的数据分布还是来自于一个生成模型。通过这两个内部模型之间不断的竞争，从而提高两个模型的生成能力和判别能力。

详细实现过程

- 假设我们现在的数据集是手写体数字的数据集minst。

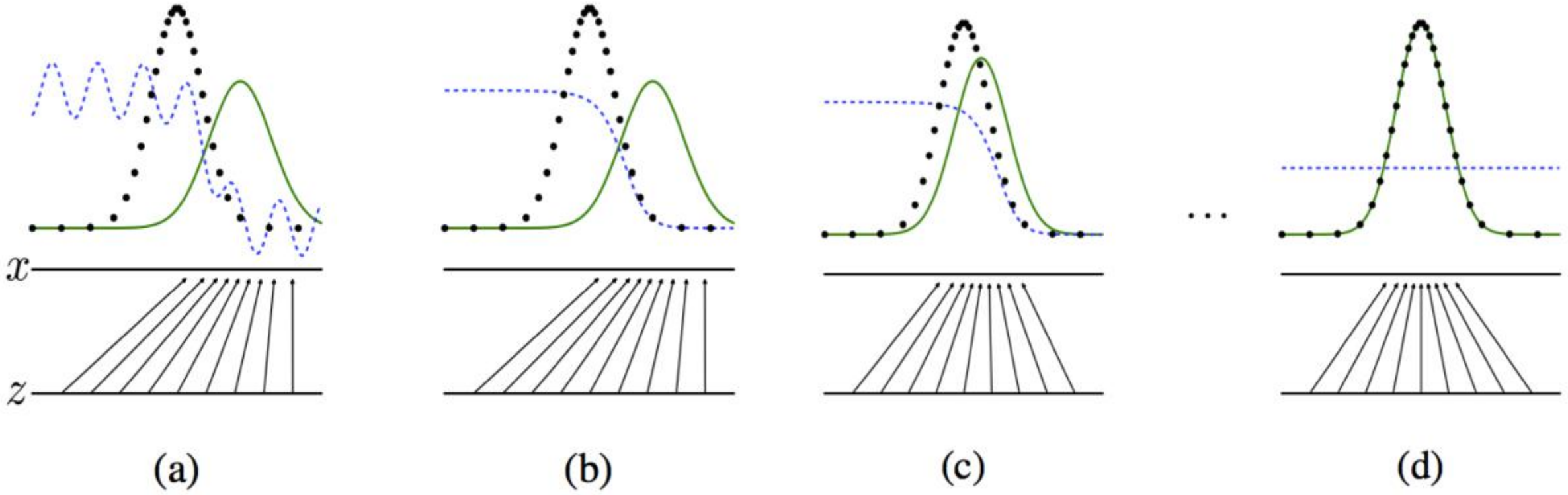
  初始化生成模型G、判别模型D（假设生成模型是一个简单的RBF，判别模型是一个简单的全连接网络，后面连接一层softmax）这些都是假设，对抗网络的生成模型和判别模型没有任何限制。



例子与训练

- 假设有一种概率分布M，它相对于我们是一个黑盒子。为了了解这个黑盒子中的东西是什么，我们构建了两个东西G和D，G是另一种我们完全知道的概率分布，D用来区分一个事件是由黑盒子中那个不知道的东西产生的还是由我们自己设的G产生的。
- 不断的调整G和D，直到D不能把事件区分出来为止。在调整过程中，需要：
- 优化G，使它尽可能的让D混淆。
- 优化D，使它尽可能的能区分出假冒的东西。
- 当D无法区分出事件的来源的时候，可以认为，G和M是一样的。从而，我们就了解到了黑盒子中的东西。

例子与训练



图a,b,c,d. 黑色的点状线代表M所产生的一些数据，绿色的线代表我们自己模拟的分布G，蓝色的线代表着分类模型D。
a图表示初始状态，b图表示，保持G不动，优化D，直到分类的准确率最高。
c图表示保持D不动，优化G，直到混淆程度最高。d图表示，多次迭代后，终于使得G能够完全你和M产生的数据，从而认为，G就是M。

博弈

- 生成式对抗网络的优化是一个二元极小极大博弈（minimax two-player game）问题，它的目的是使生成模型的输出再输入给判别模型时，判别模型很难判断是真实数据还是虚假数据。
- 训练好的生成模型，有能力把一个噪声向量转化成和训练类似的样本。

极大极小值算法

- MiniMax算法(极大极小值算法)是一种找出失败的最大可能性中的最小值的算法(即最小化对手的最大得益)，该算法通常是通过递归的形式来实现的；MiniMax算法常用于棋类等两方较量的游戏或者程序中。
- 该算法是一个零总和算法，即一方要在可选的选项中选择将其优势最大化的选择，另一方则选择令对手优势最小化的一个，其输赢的总和为0（有点像能量守恒，就像本身两个玩家都有1点，最后输家要将他的1点给赢家，但整体上还是总共有2点）。
- 由于是递归的操作，所以层次深度会非常深，那么可能使用神经网络优化

纳什均衡

- 纳什均衡是指博弈中这样的局面，对于每个参与者来说，只要其他人不改变策略，他就无法改善自己的状况。
- 纳什证明了在每个参与者都只有有限种策略选择并允许混合策略的前提下，纳什均衡定存在。
- 以两家公司的价格大战为例，价格大战存在着两败俱伤的可能，在对方不改变价格的条件下既不能提价，否则会进一步丧失市场;也不能降价,因为会出现赔本甩卖。于是两家公司可以改变原先的利益格局，通过谈判寻求新的利益评估分摊方案。
- 相互作用的经济主体假定其他主体所选择的战略为既定时，选择自己的最优战略的状态，也就是纳什均衡。

纳什均衡分类

- 纯战略纳什均衡是提供给玩家要如何进行赛局的一个完整的定义。特别地是，纯战略决定在任何一种情况下要做的移动。战略集合是由玩家能够施行的纯战略所组成的集合。而混合战略是对每个纯战略分配一个机率而形成的战略。
- 混合战略纳什均衡是允许玩家随机选择一个纯战略。混合战略博弈均衡中要用概率计算，因为每一种策略都是随机的，达到某一概率时，可以实现支付最优。因为机率是连续的，所以即使战略集合是有限的，也会有无限多个混合战略。

纳什均衡经典哲学案例——囚徒困境

- 假设有两个小偷A和B联合犯事、私入民宅被警察抓住。警方将两人分别置于不同的两个房间内进行审讯，对每一个犯罪嫌疑人，警方给出的政策是：如果一个犯罪嫌疑人坦白了罪行，交出了赃物，并且证据确凿，两人都被判有罪。如果另一个犯罪嫌疑人也作了坦白，则两人各被判刑8年；如果另一个犯罪嫌人没有坦白而是抵赖，则以妨碍公务罪（因已有证据表明其有罪）再加刑2年，而坦白者有功被减刑8年，立即释放。如果两人都抵赖，则警方因证据不足不能判两人的偷窃罪，但可以私入民宅的罪名将两人各判入狱1年。

A＼B	坦白	抵赖
坦白	-8，-8	0，-10
抵赖	-10，0	-1，-1

纳什均衡经典哲学案例——硬币正反

- 你正在图书馆枯坐，一位陌生美女主动过来和你搭讪，并要求和你一起玩个数学游戏。美女提议：“让我们各自亮出硬币的一面，或正或反。如果我们都是正面，那么我给你3元，如果我们都是反面，我给你1元，剩下的情况你给我2元就可以了。”那么该不该和这位姑娘玩这个游戏呢？这基本是废话，当然该。问题是，这个游戏公平吗？
- 每一种游戏依具其规则的不同会存在两种纳什均衡，一种是纯策略纳什均衡，也就是说玩家都能够采取固定的策略(比如一直出正面或者一直出反面)，使得每人都赚得最多或亏得最少；或者是混合策略纳什均衡，而在这个游戏中，便应该采用混合策略纳什均衡。

你\美女	美女出正面	美女出反面
你出正面	+3，-3	-2，+2
你出反面	-2，+2	+1，-1

纳什均衡经典哲学案例——枪手博弈

- 彼此痛恨的甲、乙、丙三个枪手准备决斗。甲枪法最好，十发八中；乙枪法次之，十发六中；丙枪法最差，十发四中。
- 问题1：如果三人同时开枪，并且每人只发一枪；第一轮枪战后，谁活下来的机会大一些?
- 能力差的人在竞争中耍弄手腕能赢一时，但最终往往不能成事。
- 问题2：假定甲乙丙不是同时开枪，而是他们轮流开一枪。
- 人们在博弈中能否获胜，不单纯取决于他们的实力，更重要的是取决于博弈方实力对比所形成的关系。
- **同等案例：赤壁之战、蒙古联合南宋灭金**

纳什均衡经典哲学案例——智猪博弈

- 猪圈里有两头猪，一头大猪，一头小猪。猪圈的一边有个踏板，每踩一下踏板，在远离踏板的猪圈的另一边的投食口就会落下少量的食物。如果有一只猪去踩踏板，另一只猪就有机会抢先吃到另一边落下的食物。当小猪踩动踏板时，大猪会在小猪跑到食槽之前刚好吃光所有的食物；若是大猪踩动了踏板，则还有机会在小猪吃完落下的食物之前跑到食槽，争吃到另一半残羹。

小猪\大猪	踩踏板	不踩踏板
踩踏板	0,6	-1,7
不踩踏板	3,3	0,0

- 群体活动的最大受益者“小猪”们则永远躲在幕后。活动成功了，他们可以毫发无伤地优先分到一杯羹；如果失败了，他们也可以发表一通与我无关，我是受害者之类的演讲，让“大猪”成为永远的牺牲者。从另一个角度来看，懂得智猪博弈对于个人并非是件坏事。

纳什均衡扩展案例——为什么鲜花总是插在牛粪上

- 我们把男生女生分成ABCD四等来看，由于男性的控制性倾向，导致其一般会降格选择异性伙伴，因此现实中的典型配对是：A男配B女，B男配C女，C男配D女，A女与D男轮空。

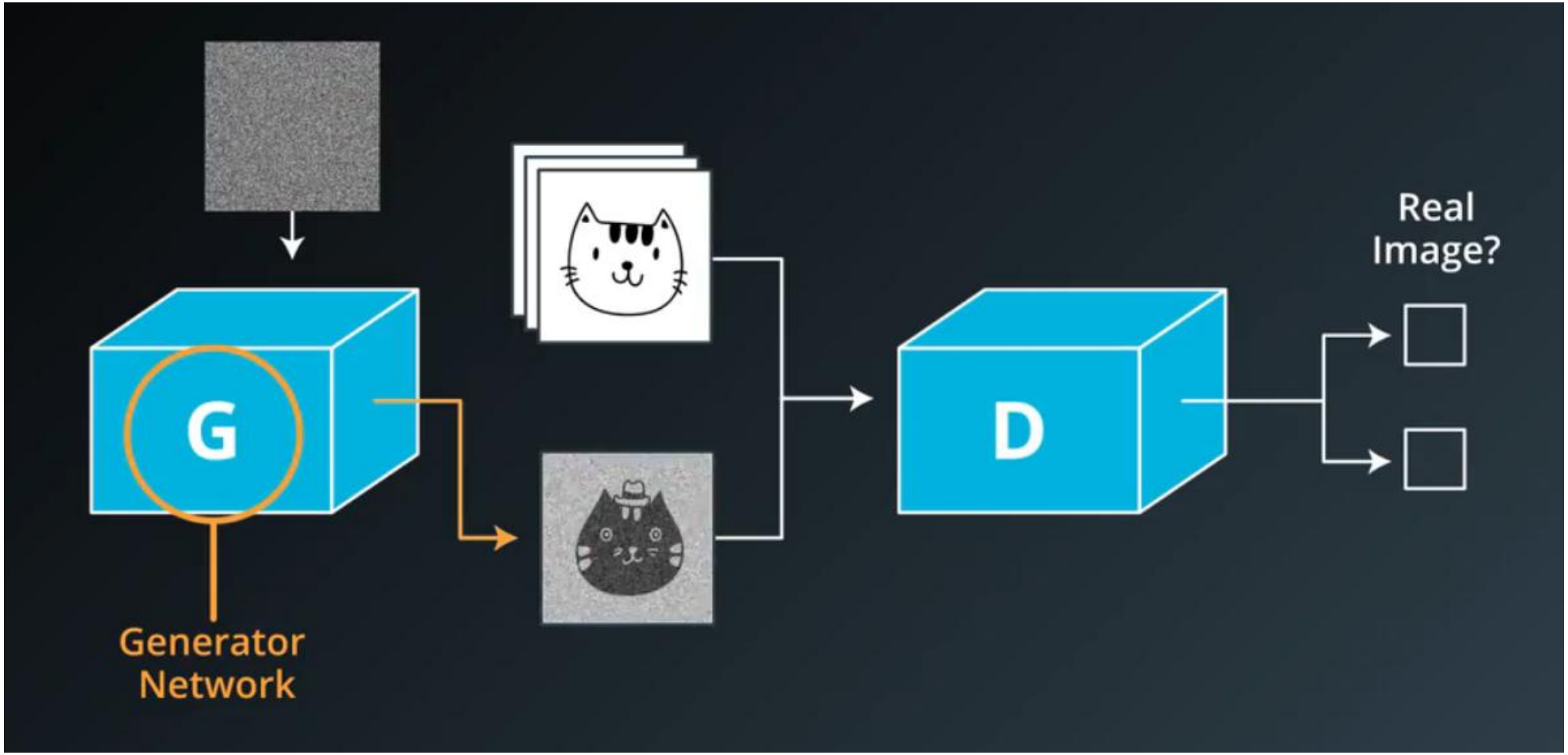
- A女(鲜花)确定D男(牛粪)没人要，而D男确定A女追不到。
- 这导致了两个最有可能的均衡策略：
  - A女如果在某种情况下选择了D男，则D男一定会接受；
  - D男去追A女则肯定不会有结果，但反正D男也没人要，追或不追A女都不会有损失，所以D男出于无聊或其他动机仍非常有可能追A女。

- **屌丝”们不要伤心，梦想还是要有的，万一哪天就“逆袭”了呢~：）**

前向传播阶段

- 可以有两种输入
  - 1、我们随机产生一个随机向量作为生成模型的数据，然后经过生成模型后产生一个新的向量，作为Fake Image，记作D(z)。
  - 2、从数据集中随机选择一张图片，将图片转化成向量，作为Real Image,记作x。
- 将由1或者2产生的输出，作为判别网络的输入，经过判别网络后输入值为一个0到1之间的数，用于表示输入图片为Real Image的概率，real为1，fake为0。

神经网络之结构



判别模型的损失函数

- 当输入的是从数据集中取出的real Iamge 数据时，我们只需要考虑第二部分，D(x)为判别模型的输出，表示输入x为real 数据的概率，我们的目的是让判别模型的输出D（x）的输出尽量靠近1。
- 当输入的为fake数据时，我们只计算第一部分，G（z）是生成模型的输出，输出的是一张Fake Image。我们要做的是让D(G(z))的输出尽可能趋向于0。这样才能表示判别模型是有区分力的。
- **相对判别模型来说，这个损失函数其实就是交叉熵损失函数。计算loss，进行梯度反传。**这里的梯度反传可以使用任何一种梯度修正的方法。 当更新完判别模型的参数后，我们再去更新生成模型的参数。

$$-((1-y)\log(1-D(G(z)))+y\log D(x))$$

生成模型的损失函数

- 对于生成模型来说，我们要做的是让G（z）产生的数据尽可能的和数据集中的数据一样。就是所谓的同样的数据分布。那么我们要做的就是最小化生成模型的误差，即只将由G（z）产生的误差传给生成模型。

$$(1-y)\log(1-D(G(z)))$$

- 但是针对判别模型的预测结果，要对梯度变化的方向进行改变。当判别模型认为G（z）输出为真实数据集的时候和认为输出为噪声数据的时候，梯度更新方向要进行改变。
- 其中$\overline{D}$ 表示判别模型的预测类别，对预测概率取整，为0或者1。用于更改梯度方向，阈值可以自己设置，或者正常的话就是0.5。

$$(1-y)\log(1-D(G(z)))(2*\overline{D}(G(z))-1)$$

判别模型的目标函数

- 用数学语言描述整个博弈过程的话，就是：假设我们的生成模型是g(z)，其中z是一个随机噪声，而g将这个随机噪声转化为数据类型x，仍拿图片问题举例，这里g的输出就是一张图片。D是一个判别模型，对任何输入x，D(x)的输出是0-1范围内的一个实数，用来判断这个图片是一个真实图片的概率是多大。令Pr和Pg分别代表真实图像的分布与生成图像的分布。

$$\max_D E_{x\sim P_r}[\log D(x)] + E_{x\sim P_g}[\log(1 - D(x)]$$

整体目标函数

- 将上述例子所描述的过程公式化，得到如上公式。公式中D(x)表示x属于分布M的概率，因而，优化D的时候就是让V(D,G)最大，优化G的时候就是让V(D,G)最小。其中，x~pdata(x) 表示x取自真正的分布。 z~pz(z) 表示z取自我们模拟的分布。G表示生成模型，D表示分类模型。

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

整体目标函数

- 在我们的函数V(D,G)中，第一项是来自实际分布（pdata（x））的数据通过鉴别器（也称为最佳情况）的熵（Entropy）。鉴别器试图将其最大化为1。第二项是来自随机输入（p（z））的数据通过发生器的熵。生成器产生一个假样本， 通过鉴别器识别虚假（也称为最坏的情况）。在这一项中，鉴别器尝试将其最大化为0（即生成的数据是伪造的的概率的对数是0）。所以总体而言，鉴别器正在尝试最大化函数V(D,G)。
- 另一方面，生成器的任务完全相反，它试图最小化函数V(D,G)，使真实数据和假数据之间的区别最小化。这就是说，生成器和鉴别器像在玩猫和老鼠的游戏。

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

算法流程图

- 下图是原文给的算法流程，noise 就是随机输入生成模型的值。

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

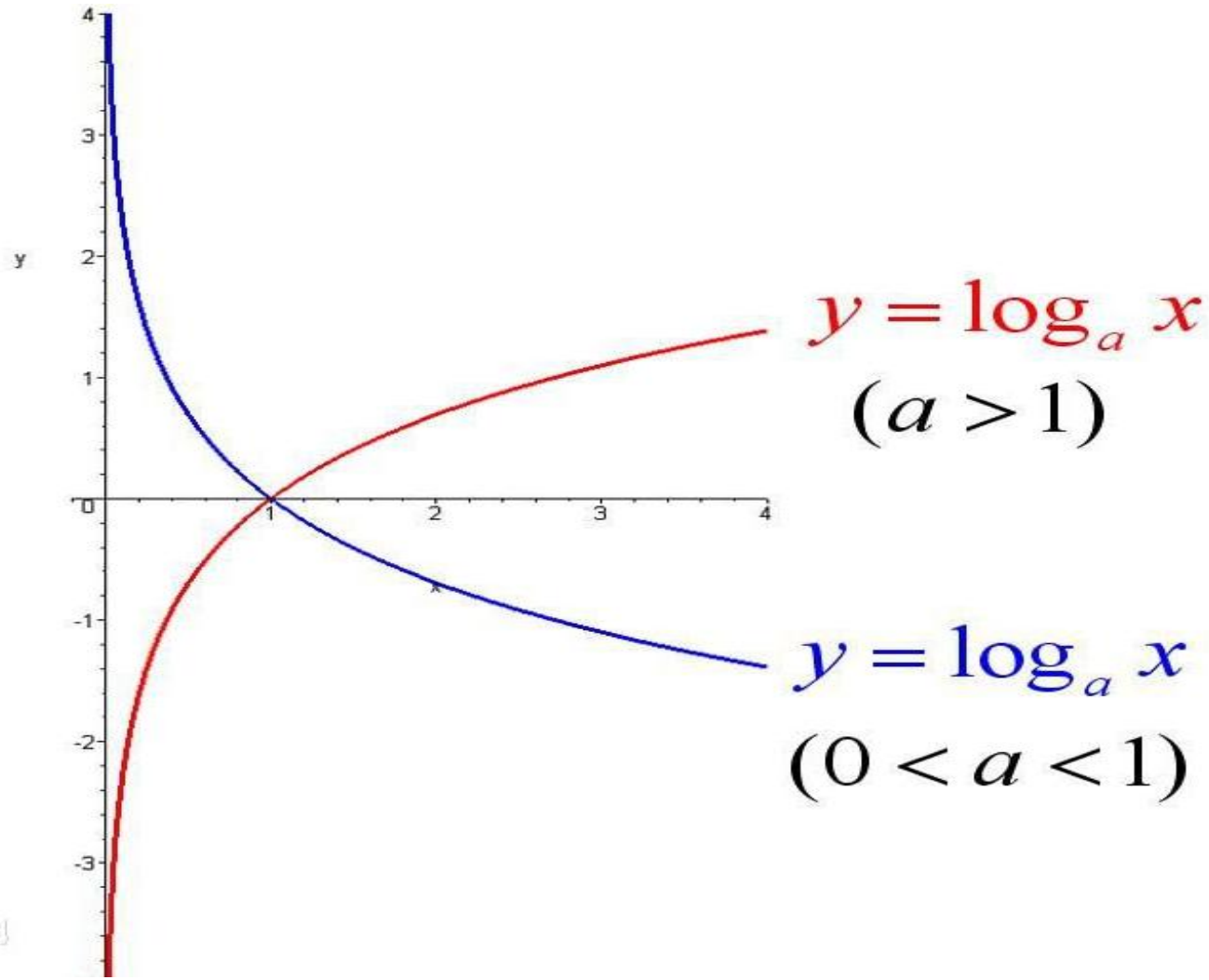
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

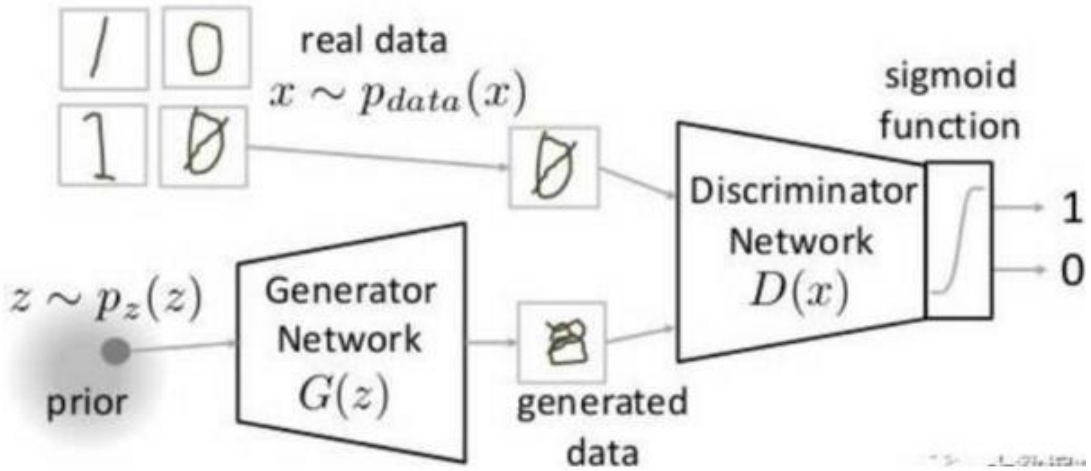
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

算法流程图



训练细节

- 训练阶段包括顺序完成的两个阶段
- **第一阶段：**训练鉴别器/判别器，冻结生成器（冻结意思是不训练，神经网络只向前传播，不进行Backpropagation反向传播）
- **第二阶段：**训练生成器，冻结鉴别器/判别器。



训练细节

- **训练GAN的步骤**
  - **第1步**：定义问题。你想生成假的图像还是文字？你需要完全定义问题并收集数据。
  - **第2步**：定义GAN的架构。GAN看起来是怎么样的，生成器和鉴别器应该是多层感知器还是卷积神经网络？这一步取决于你要解决的问题。
  - **第3步**：用真实数据训练鉴别器N个epoch。训练鉴别器正确预测真实数据为真。这里N可以设置为1到无穷大之间的任意自然数。
  - **第4步**：用生成器产生假的输入数据，用来训练鉴别器。训练鉴别器正确预测假的数据为假。
  - **第5步**：用鉴别器的出入训练生成器。当鉴别器被训练后，将其预测值作为标记来训练生成器。训练生成器来迷惑鉴别器。
  - **第6步**：重复第3到第5步多个epoch
  - **第7步**：手动检查假数据是否合理。如果看起来合适就停止训练，否则回到第3步。这是一个手动任务，手动评估数据是检查其假冒程度的最佳方式。当这个步骤结束时，就可以评估GAN是否表现良好。

证明

- 当G固定的时候，D会有唯一的最优解。证明一如下：

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

证明如下：

- 首先，对V(G,D)进行变换

$$\begin{aligned} V(G,D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned}$$

- 对于任意的a,b $\in R^2 \setminus \{0, 0\}$, 下面的式子在a/(a+b)处达到最优。

$$y \rightarrow a\log(y) + b\log(1 - y)$$

证明

- 根据证明一，可以对V(G,D)中最大化D的步骤进行变换。

$$
\begin{aligned}
C(G) &= \max_D V(G,D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

- 从而得到定理

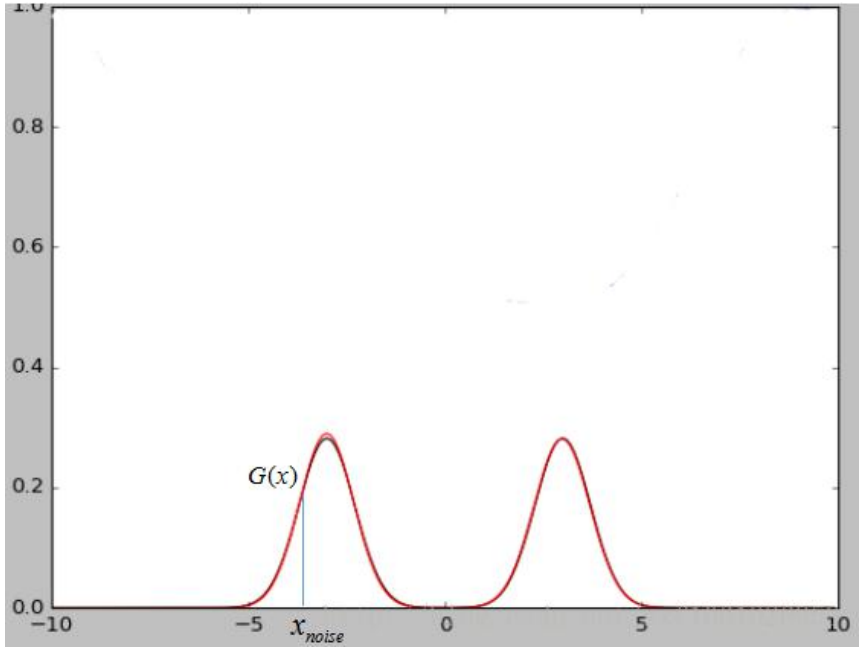
Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

直接带入$p_g=p_{data}$可得-log4，当入$p_g!=p_{data}$时，得到

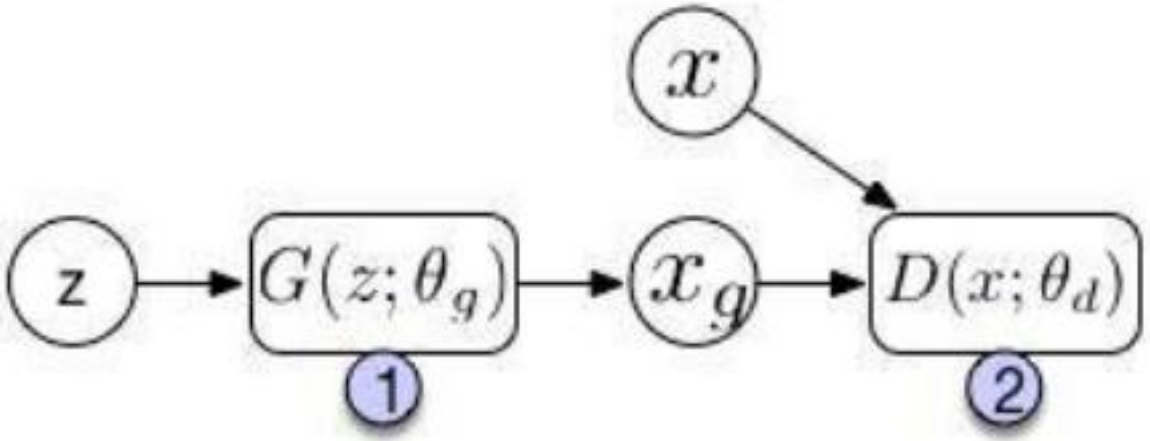
$$
C(G) = -\log(4) + KL\left(p_{\text{data}} \,\middle\|\, \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \,\middle\|\, \frac{p_{\text{data}} + p_g}{2}\right)
$$

noise输入

- 假设我们现在的数据集是一个二维的高斯混合模型，那么这么noise就是x轴上我们随机输入的点，经过生成模型映射可以将x轴上的点映射到高斯混合模型上的点。当我们的数据集是图片的时候，那么我们输入的随机噪声其实就是相当于低维的数据，经过生成模型G的映射就变成了一张生成的图片G（x）。
- 最终两个模型达到稳态的时候判别模型D的输出接近1/2，也就是说判别器很难判断出图片是真是假，这也说明了网络是会达到收敛的。



GANS总结



$$\min_G \{\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]\}$$

3

GANS总结

- 图中上半部分是GAN模型的基本架构。我们先从一个简单的分布中采样一个噪声信号 z（实际中可以采用[0, 1]的均匀分布或者是标准正态分布），然后经过一个生成函数后映射为我们想要的数据分布 Xg （z 和 X 都是向量）。生成的数据和真实数据都会输入一个识别网络 D。识别网络通过判别，输出一个标量，表示数据来自真实数据的概率。
- 在实现上，G 和 D 都是可微分函数，都可以用多层神经网络实现。因此上面的整个模型的参数就可以利用backpropagation来训练得到。
- 图中的下半部分是模型训练中的目标函数。仔细看可以发现这个公式很像cross entropy，注意D是 P(Xdata) 的近似。对于 D 而言要尽量使公式最大化（识别能力强），而对于 G 又想使之最小（生成的数据接近实际数据）。
- 整个训练是一个迭代过程，但是在迭代中，对 D 的优化又是内循环。所以每次迭代，D 先训练 k次，G 训练一次。

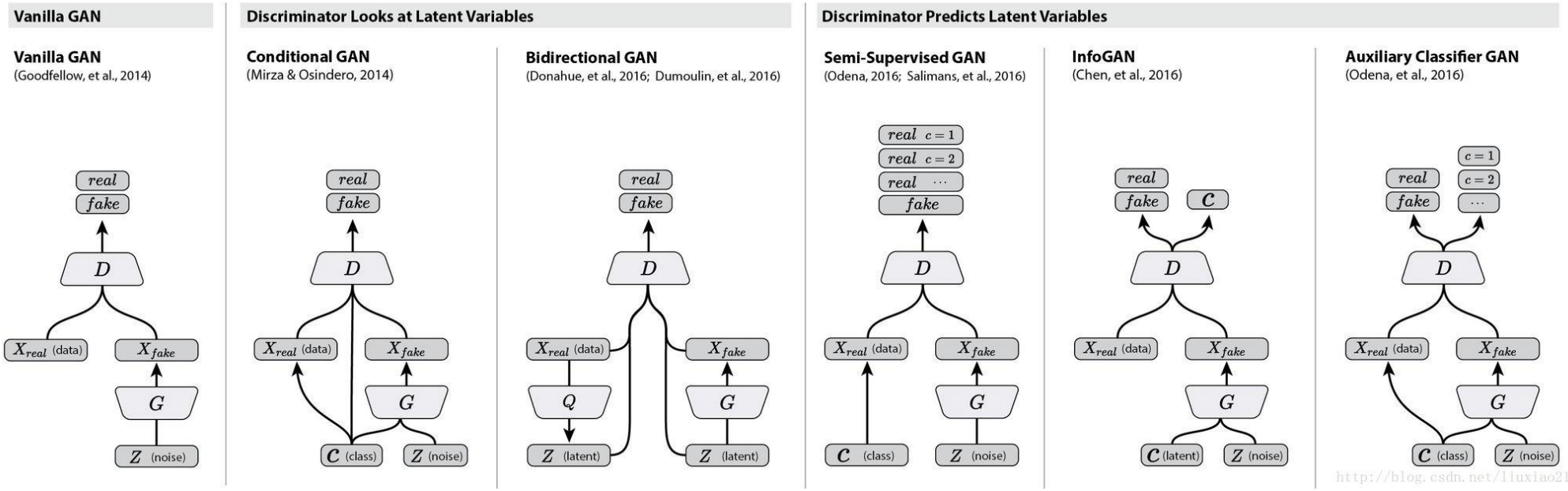
优势和劣势

- 优势：
  - Markov链不需要了，只需要后向传播就可以了。
  - 生成网络不需要直接用样本来更新了，这是一个可能存在的优势。
  - 对抗网络的表达能力更强劲，而基于Markov链的模型需要分布比较模糊才能在不同的模式间混合。

- 劣势：
  - 对于生成模型，没有直接的表达，而是由一些参数控制。
  - D需要和G同步的很好才可以。

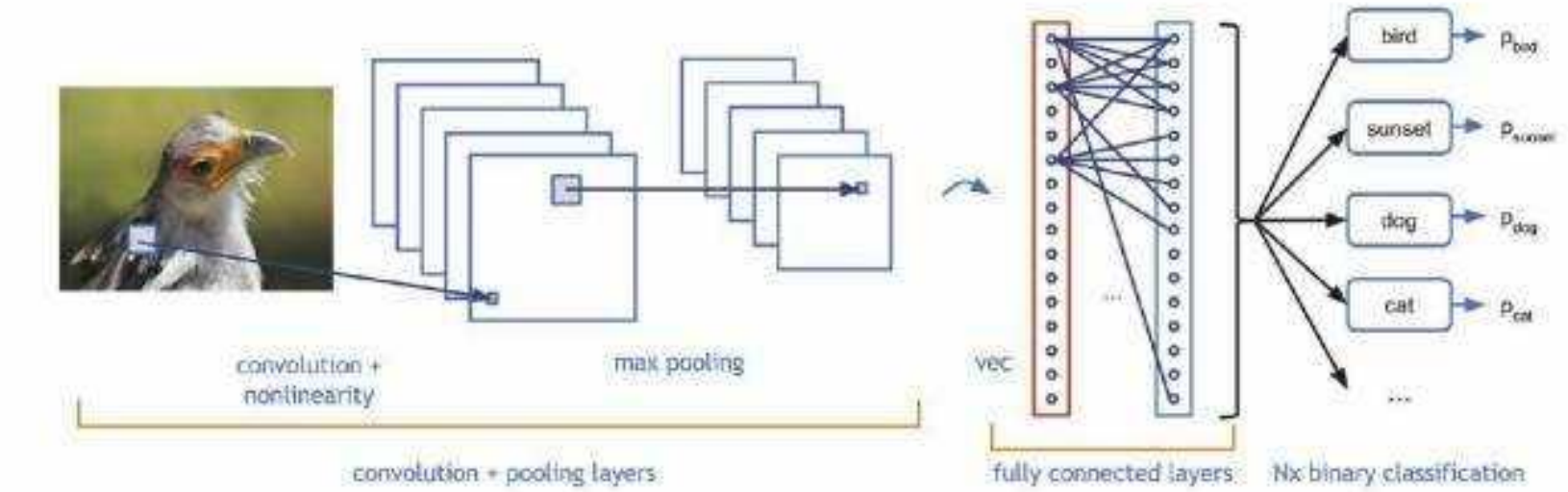
拓展延伸

- GAN模型最大的优势就是训练简单，但是也有缺点比如训练的稳定性。
- 有趣的是，在论文Generative Adversarial Nets的future work部分，作者提出了5个可能扩展的方向，而现在回过头来看，后续的很多工作真的就是在照着这几个思路填坑。



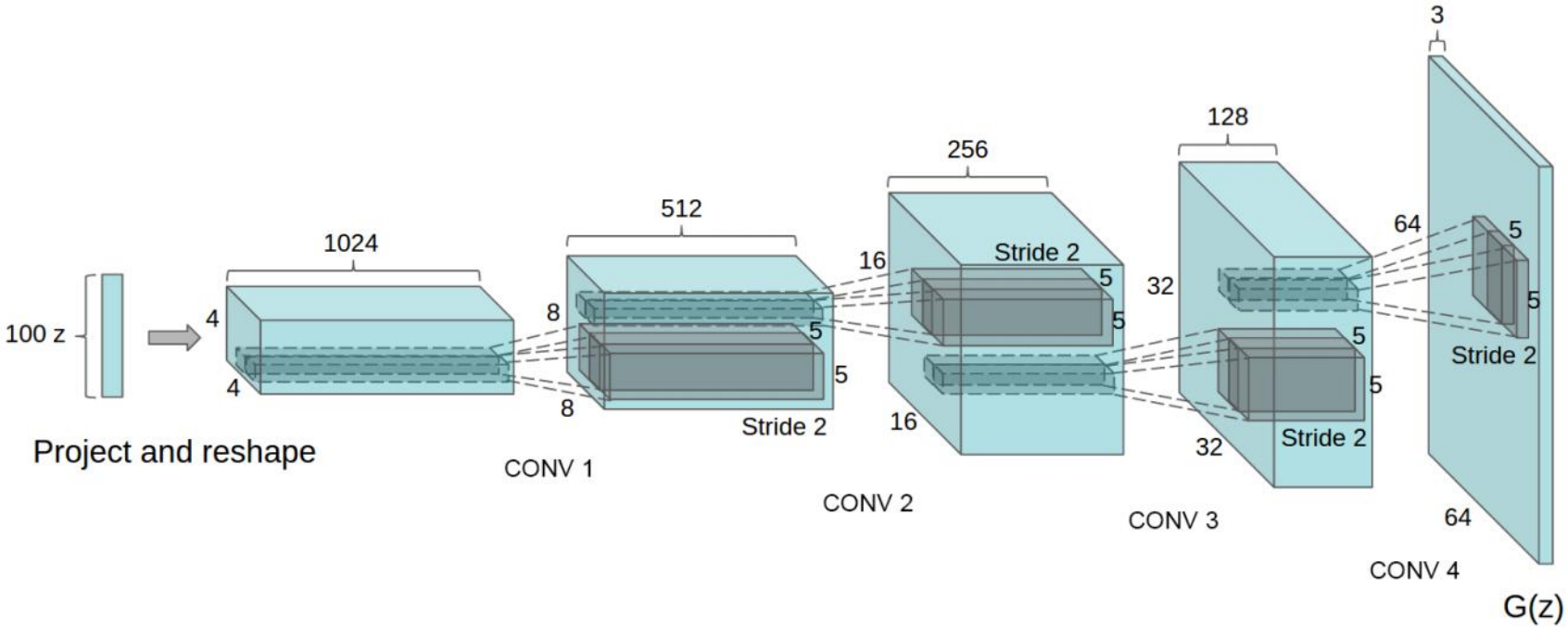
DCGAN

- 在图像生成过程中，如何设计生成模型和判别模型呢？深度学习里，对图像分类建模，刻画图像不同层次，抽象信息表达的最有效的模型是：CNN（convolutional neural network，卷积神经网络）。



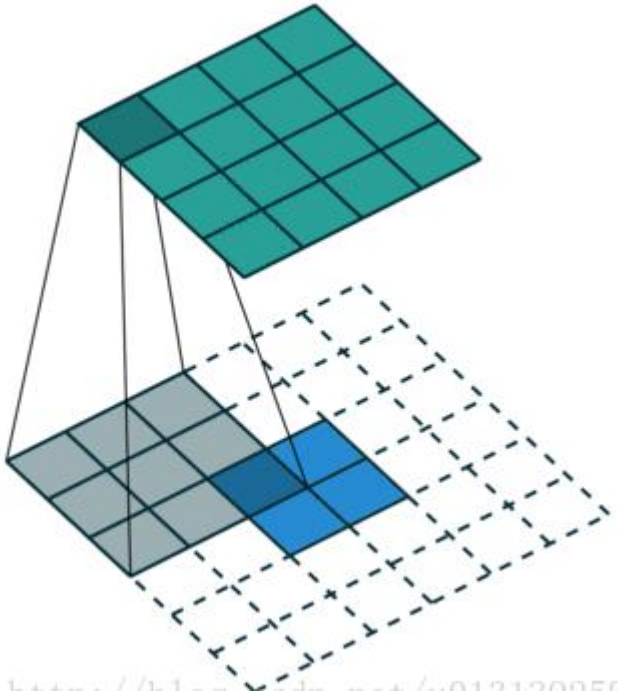
DCGAN

- 那么生成图像的模型应该是什么样子的呢？想想小时候上美术课，我们会先考虑构图，再勾画轮廓，然后再画细节，最后填充颜色，这事实上也是一个多层级的过程，就像是把图像理解的过程反过来，于是，人们为图像生成设计了一种类似反卷积的结构：Deep convolutional NN for GAN（DCGAN）



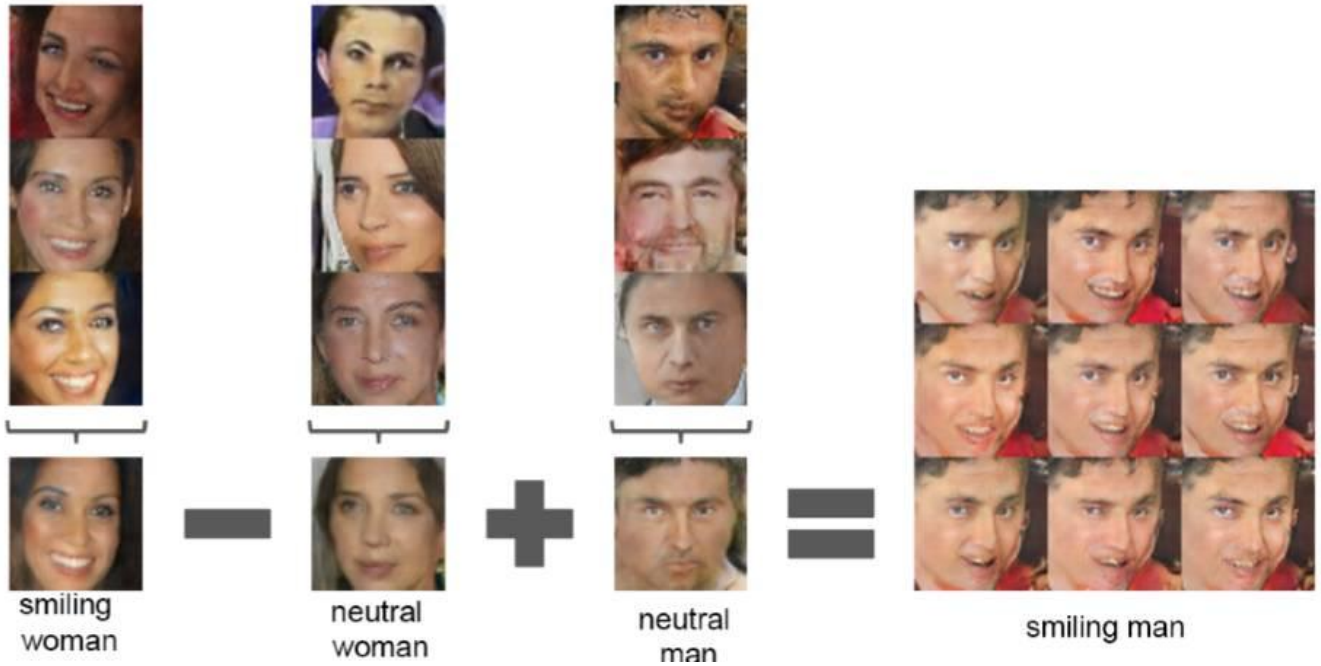
DCGAN

- 2x2的输入信号，经过3x3 的filters，产生了4x4的feature map。从小的维度产生大的维度，所以transport-convolution又称为上采样卷积。



DCGAN

- DCGAN采用一个随机噪声向量作为输入，如高斯噪声。
- 输入通过与CNN类似但是相反的结构，将输入放大成二维数据。
- 通过采用这种结构的生成模型和CNN结构的判别模型，DCGAN在图片生成上可以达到相当可观的效果。



DCGAN训练稳定原因

- 使用步长卷积代替上采样层，卷积在提取图像特征上具有很好的作用，并且使用卷积代替全连接层。
- 生成器G和判别器D中几乎每一层都使用batchnorm层，将特征层的输出归一化到一起，加速了训练，提升了训练的稳定性。（生成器的最后一层和判别器的第一层不加batchnorm）
- 在判别器中使用leakrelu激活函数，而不是RELU，防止梯度稀疏，生成器中仍然采用relu，但是输出层采用tanh
- 使用adam优化器训练，并且学习率最好是0.0002。

GAN的改进——WGAN

- 刚才谈到很多GAN的优点、应用和变种，那么GAN真的是完美无缺的吗?
- 其实使用过GAN的人应该知道，训练GAN有很多头疼的问题。例如：GAN的训练对超参数特别敏感，需要精心设计。GAN中关于生成模型和判别模型的迭代也很有问题，按照通常理解，如果判别模型训练地很好，应该对生成的提高有很大作用，但实际中恰恰相反，如果将判别模型训练地很充分，生成模型甚至会变差。那么问题出在哪里呢?

GAN的改进——WGAN

- 在ICLR 2017大会上有一篇口头报告论文提出了这个问题产生的机理和解决办法。问题就出在目标函数的设计上。这篇文章的作者证明，GAN的本质其实是优化真实样本分布和生成样本分布之间的差异，并最小化这个差异。特别需要指出的是，优化的目标函数是两个分布上的Jensen-Shannon距离，但这个距离有这样一个问题，如果两个分布的样本空间并不完全重合，这个距离是无法定义的。
- 作者接着证明了“真实分布与生成分布的样本空间并不完全重合”是一个极大概率事件，并证明在一些假设条件下，可以从理论层面推导出一些实际中遇到的现象。
- 解决方案：使用Wasserstein距离代替Jensen-Shannon距离。并依据Wasserstein距离设计了相应的算法，即WGAN。新的算法与原始GAN相比，参数更加不敏感，训练过程更加平滑。

WGAN对GAN的改进

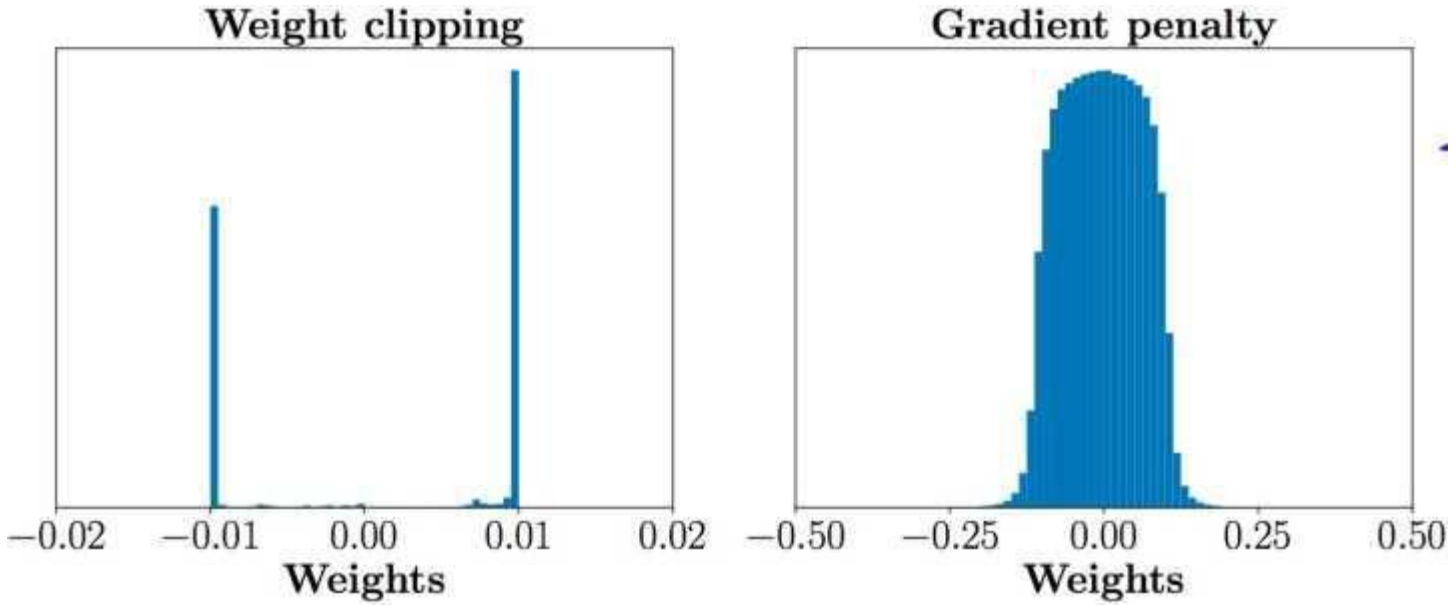
- 判别器最后一层去掉sigmoid
- 生成器和判别器的loss不取log
- 对更新后的权重强制截断到一定范围内，比如[-0.01，0.01]，以满足论文中提到的lipschitz连续性条件。
- 论文中也推荐使用SGD， RMSprop等优化器，不要基于使用动量的优化算法，比如adam。不过实际使用adam也是可以的，只是效果不佳。

WGAN总结

- 总的来说，GAN中交叉熵（JS散度）不适合衡量生成数据分布和真实数据分布的距离，如果通过优化JS散度训练GAN会导致找不到正确的优化目标，所以，WGAN提出使用wassertein距离作为优化方式训练GAN。
- 但是数学上和真正代码实现上还是有区别的，使用Wasserteion距离需要满足很强的连续性条件—lipschitz连续性，为了满足这个条件，使用了将权重限制到一个范围的方式强制满足lipschitz连续性，但是这也造成了隐患。不过，虽然理论证明很漂亮，但是实际上训练起来，以及生成结果并没有期待的那么好。

WGAN-GP

- WGAN-GP是WGAN之后的改进版，主要还是改进了连续性限制的条件，因为，作者也发现将权重剪切到一定范围之后，比如剪切到[-0.01,+0.01]后，发生了这样的情况，如下图左边表示。



WGAN-GP优化

- 大多数的权重都在-0.01 和0.01上，这就意味了网络的大部分权重只有两个可能数，对于深度神经网络来说不能充分发挥深度神经网络的拟合能力，简直是极大的浪费。并且强制剪切权重容易导致梯度消失或者梯度爆炸 。
- 为了解决这个问题，并且找一个合适的方式满足lipschitz连续性条件。WGAN-GP使用梯度惩罚（gradient penalty）的方式以满足此连续性条件，其结果如上图右边所示。

梯度惩罚

- 梯度惩罚就是既然Lipschitz限制是要求判别器的梯度不超过K，那么可以通过建立一个损失函数来满足这个要求，即先求出判别器的梯度d(D(x))，然后建立与K之间的二范数就可以实现一个简单的损失函数设计。
- 作者提出没必要对整个数据集（真的和生成的）做采样，只要从每一批次的样本中采样就可以了，比如可以产生一个随机数，在生成数据和真实数据上做一个插值。

$$X' = X * \theta + (1 - \theta) * X_{fake}$$

WGAN-GP的贡献

- 提出了一种新的lipschitz连续性限制手法—梯度惩罚，解决了训练梯度消失梯度爆炸的问题。
- 比标准WGAN拥有更快的收敛速度，并能生成更高质量的样本。
- 提供稳定的GAN训练方式，几乎不需要怎么调参，成功训练多种针对图片生成和语言模型的GAN架构。

LSGAN

- 即使用了最小二乘损失函数代替了GAN的损失函数。
- 但是就这样的改变，缓解了GAN训练不稳定和生成图像质量差多样性不足的问题。
- 事实上，使用JS散度并不能拉近真实分布和生成分布之间的距离，使用最小二乘可以将图像的分布尽可能的接近决策边界。

$$\min_D V_{\mathrm{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-a)^2\right]$$

$$\min_G V_{\mathrm{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-c)^2\right],$$

BEGAN

- BEGAN的主要原理，BEGAN和其他GAN不一样，这里的D使用的是auto-encoder（AE自编码）结构，就是下面这种，D的输入是图片，输出是经过编码解码后的图片。
- 以往的GAN以及其变种都是希望生成器生成的数据分布尽可能的接近真实数据的分布，当生成数据分布等同于真实数据分布时，我们就确定生成器G经过训练可以生成和真实数据分布相同的样本，即获得了生成足以以假乱真数据的能力。研究者们设计了各种损失函数去令G的生成数据分布尽可能接近真实数据分布。
- BEGAN代替了这种估计概率分布方法，它不直接去估计生成分布Pg与真实分布Px的差距，进而设计合理的损失函数拉近他们之间的距离，而是估计分布的误差之间的距离，只要分布的的误差分布相近的话，也可以认为这些分布是相近的。即如果我们认为两个人非常相似，又发现这两人中的第二个人和第三个人很相似，那么我们就完全可以说第一个人和第三个人长的很像。

神经网络案例

- 生成手写数字集，代码详见：《手写数字集》
- DCGAN实现，代码详见：《DCGAN》
- 作业：生成明星脸，代码详见：《生成明星脸》