第八讲
生成模型
Lecture 8
Generative Models

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

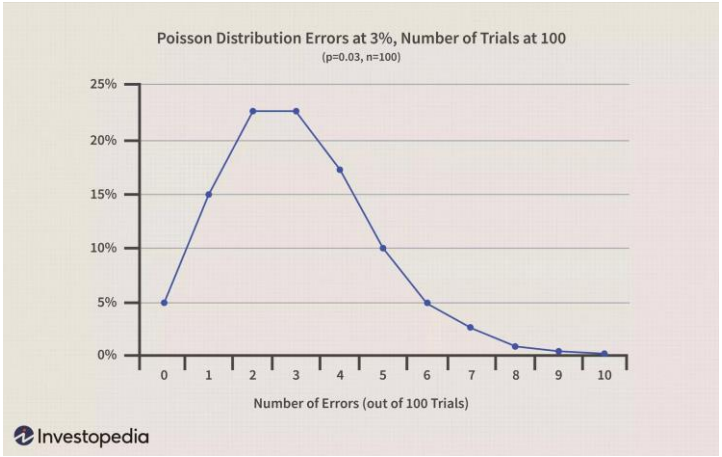
- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

采样问题
Sampling Problem

- 在实践中，我们经常需要从特定分布中采样，以进行研究或测试。比如，我们开发了一个客户服务系统，要对之进行压力测试。我们知道，一般，到店客户数量服从泊松分布。我们可从特定时段的泊松分布采样，来模拟系统需要服务的数目。

  In practice, we have to sample from a specific probability distribution to cater to some requirement, such as for research or test. For example, to exert a load test on a newly-implemented customer serving system. We know usually the number of customers visiting a complex department is in accordance with Poisson distribution. So, we can sample from an instance of Poisson distribution on specified occasions to simulate the customers to test the system.



采样问题
Sampling Problem

- 什么是采样呢？说白了，就是生成特定分布的随机数或样本。比如，生成符合均一分布的特定范围内的随机数，就是所谓的采样。

  What is sampling? In plain text, it is to generate the numbers (or samples) that in accordance with some specific distribution. For instance, if it is desired to generate numbers of uniform distribution in a given range, it is the so called sampling.

$$X_{n+1} = [a * X_n] \pmod m$$

where: $a = 16{,}807$, and $m = 2^{31} - 1$

采样问题
Sampling Problem

- 这只是一个比较简单的例子，有时我们需要从更为复杂的分布采样，这时会有两个问题：

  The above is just a simple example, sometimes, we have to sample from a much more complicated distribution,

  - 这个分布很复杂，不好直接采样；

    The distribution is overly complicated and rendering direct sampling very difficult;

  - 基于领域知识，我们只对样本分布有一个大概的了解，无法写出解析表达。

    We only have a gross understanding of the distribution; an analytic form of the distribution is intractable.

换元
Change of Variable

- 这个分布很复杂，不好直接采样；

  The distribution is overly complicated and rendering direct sampling very difficult;

- 在这种情况下，会采用换元法，求出简单变换到复杂变换下的关系式；这样可以先从简单分布中采样，再变换到复杂分布下的样本。

  In this circumstance, the change of variables will be adopted to find the relation which guides the shift from the simple distribution to complex distribution. In this way, we can sample from the simple distribution then convert it to the complicated distribution.

无解析形式密度函数问题
Intractability of PDF

- 基于领域知识，我们只对样本分布有一个大概的了解，无法写出解析表达。

  We only have a gross understanding of the distribution; an analytic form of the distribution is intractable.

- 比如，之前讲到人脸像素值的概率分布，我们知道的只是一个个样本，即生活中我们所见的人脸；并且，对于给定的一个样本，我们还指导究竟是不是人脸；至于这样的分布有怎样的一个解析表达，大概没有人知道。

  For example, with regard to the pixel value distribution of human face, actually we are just aware of the samples themselves, namely, individual faces during daily life. And further, for any presented sample, we have the knowledge to decide whether it is human face or not. However, for a potential analytic form of such a distribution, no one knows.

对抗式生成网络
Generative Adversarial Networks

- 通用逼近定理：在人工神经网络的理论研究中，有一个很著名的定理，即如果网络足够大，足够深，则其可以以任意精度逼近任意性质良好的函数。假设我们知道复杂概率分布函数的解析形式，则我们可能可以求出来这个转换函数，问题是我们不知道。但既然神经网络可以逼近任意函数，则我们可以假设神经网络亦可以逼近这个转换函数，问题是，怎样衡量这个由神经网络建模的转换函数就是我们需要的。

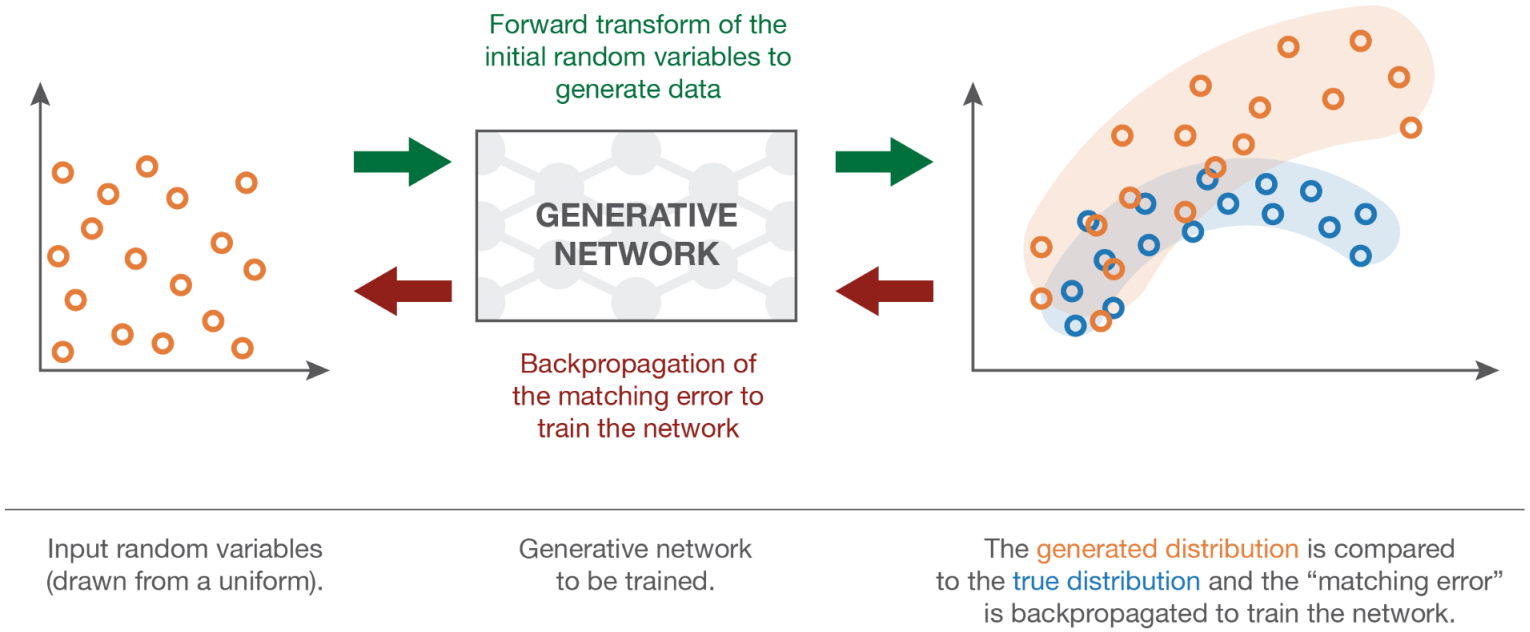
  Universal proximation theorem: during the theoretic research of neural networks, it came up the theorem that by leveraging sufficient deep and wide neural network, any complicated function with well-stated properties can be approximated in any precision. If we know the analytical form of the complicated PFD, then we can try to derive the transform function. The situation is we don’t know. But considering arbitrary functions could be approximated by neural networks, we just assume neural networks can approach such pre-exist function by conjecture. The problem is, how can we measure the function modeled by the network is all we needed.

对抗式生成网络
Generative Adversarial Networks

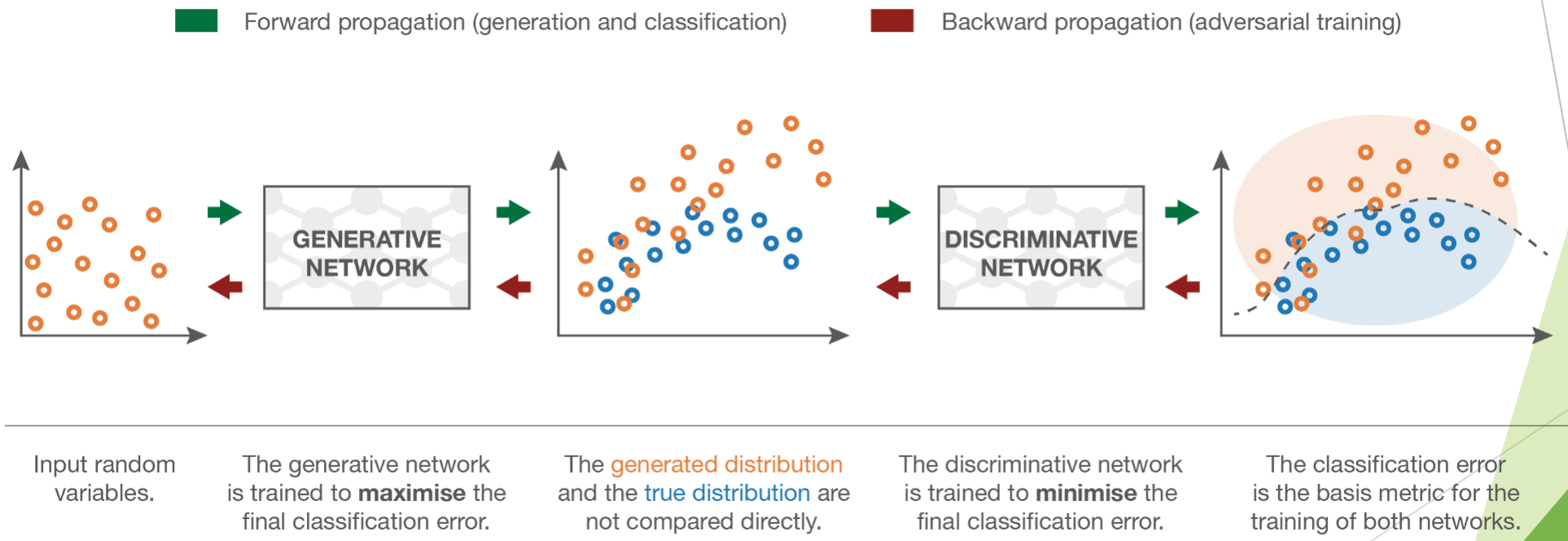
- 要度量我们建模的转换函数的质量，能利用的只有一个个具体的样本；那么怎么利用这些样本呢？一个直观的想法是，假设用网络表征的转换函数记为生成器，或$G(z)$，如果从这个函数中采出或生成的样本，与实际样本不可区分，则我们认为这个转换函数正是我们需要的。我们随后便可以从中采样，以便进行研究或测试。但人工判断这些样本的不可区分性，徒劳费神，因此我们可以再构造一个网络，称为判别器，或$D(x)$，来达到整体的目的。

  To measure the quality of the modeled transform function by utilizing the individual concrete samples, an intuitive idea is by harnessing the approximation capacity of networks, which denoted as generator or $G(z)$, to distinguish the samples from the generated and real samples. If they are indistinguishable, then such a $G(z)$ is the desired one. However, to manually identify these samples are time consuming, so, we can construct another network named discriminator or $D(x)$ for such a purpose.

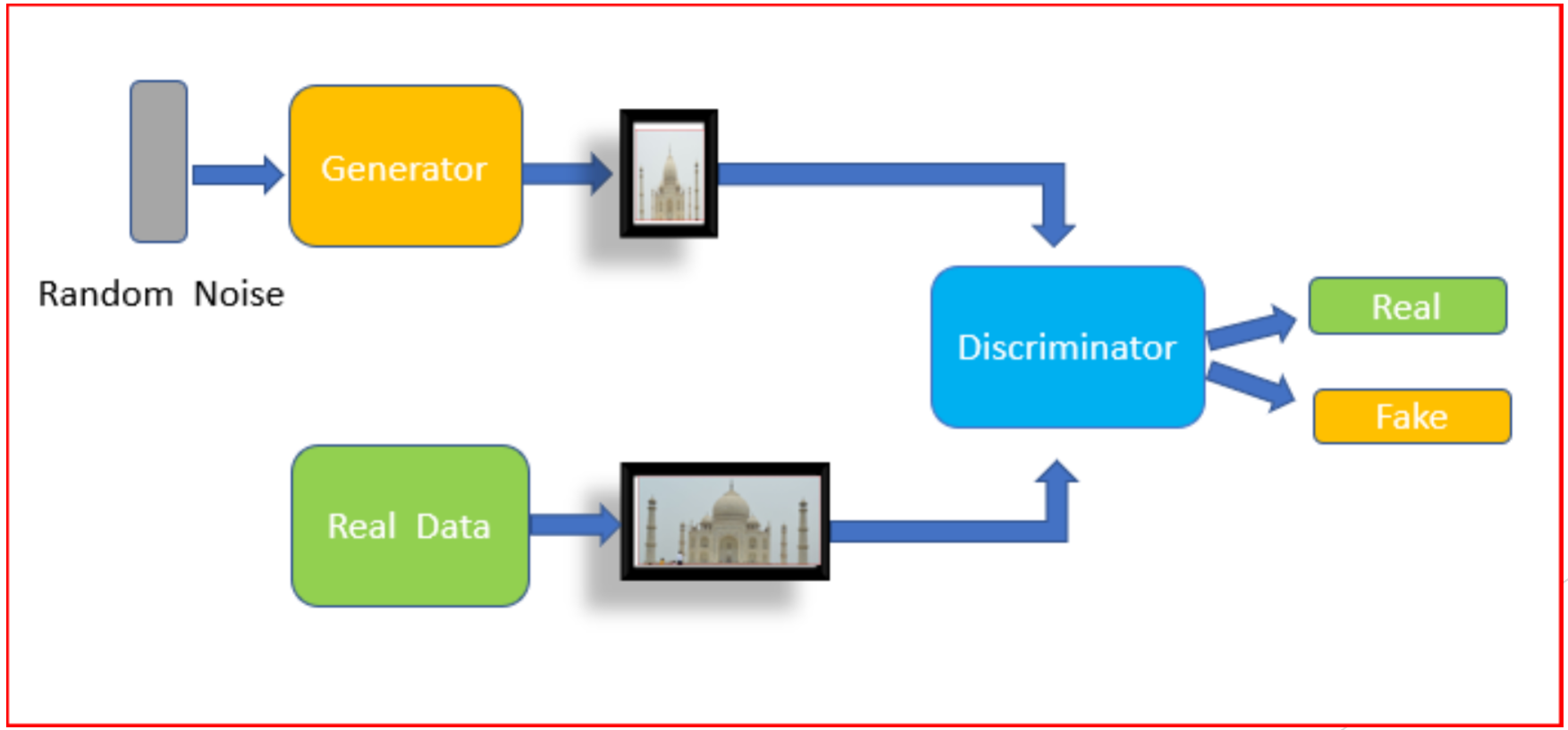
对抗式生成网络
Generative Adversarial Networks



对抗式生成网络
Generative Adversarial Networks



对抗式生成网络
Generative Adversarial Networks



对抗式生成网络
Generative Adversarial Networks

- z: 噪声，可以认为从简单分布中的随机采样；noise, can be treated as samples from simple distributions;
- x_{real}：真实图片；real images;
- x_{fake}：生成图片；generated images;
- G：生成器；generator;
- D：判别器；discriminator;
- $G(z)$ or x_{fake}：生成器输出；generator output
- $D(x)$：判别器输出；discriminator output

对抗式生成网络
Generative Adversarial Networks

- 显然，为达成所述之目的，就需要定义恰当的损失函数，以便可以用反向传播算法训练；对于判别器，我们期望其输出是判定为真实图片的概率，则可构造如下的损失函数：

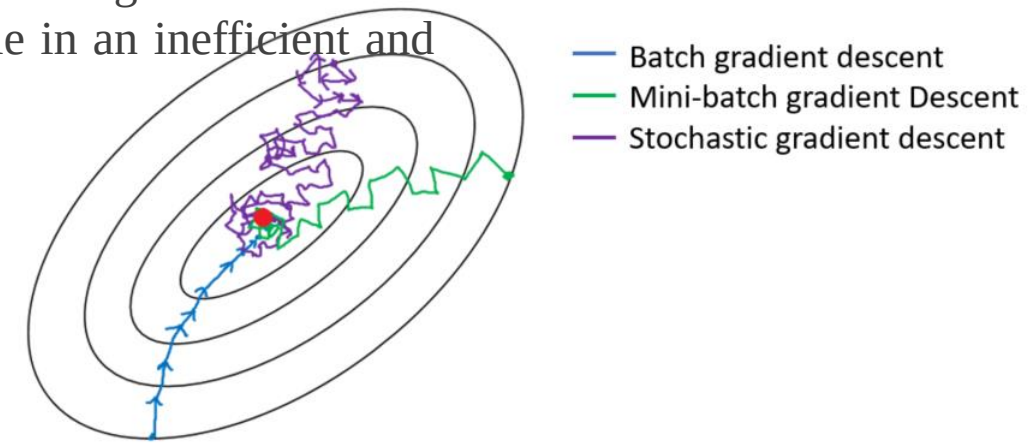
  Obviously, to fulfill the aforementioned aim, it is necessary to define the proper loss function in order to apply the back-propagation algorithm. For the discriminator, by design the output is interpreted as the probabilities of real images. So, the constructed loss function is as follows:

$$\max_D L(D) = 1 \cdot \log D(x_{real}) + 1 \cdot \log\Big(1 - D\big(G(z)\big)\Big)$$

对抗式生成网络
Generative Adversarial Networks

- 注意上述损失函数只是为了便于理解，真实场景必须批量训练。原因在于，GAN要求的是生成样本的分布与真实分布不可区别。而只有当样本足够多的时候才能谈分布，单个样本是无法谈分布的。而前面无论CNN与RNN，单个样本是可以训练的，无非就是过程曲折些。

  Please note the loss function above is just for illustrative understanding. In practice the training must be performed in the batch mode. The reason is for GAN, it only distinguishes the distributions of real samples v.s. generated samples. And the concept of distribution only holds for a collective of samples instead of a single one. But the CNN and RNN mentioned before can be trained with one sample in an inefficient and unstable process.



对抗式生成网络
Generative Adversarial Networks

- 实际可行的损失函数如下：

  In practice, the loss function is as follows:

$$\max_D L(D) = \sum_{i=1}^{m} 1 \cdot \log D\left(x_{real}^{(i)}\right) + \sum_{j=1}^{n} 1 \cdot \log\left(1 - D\left(G\left(z^{(j)}\right)\right)\right)$$

- 或写成更为紧凑的形式：

  Or re-write as a more compact form:

$$\max_D L(D) = \mathbb{E}_{x \sim p_x(x)} \log D(x) + \mathbb{E}_{z \sim p_z(x)} \log\left(1 - D\left(G(z)\right)\right)$$

对抗式生成网络
Generative Adversarial Networks

- 对于生成器，我们期望其所生成之图片与真实图片近可能接近，即被判为真实图片的概率尽可能高，则其损失函数为$L(G) = 1 \cdot \log\left(D\left(G(z)\right)\right)$尽可能大，由于先前已有$1 \cdot \log\left(1 - D\left(G(z)\right)\right)$形式，则要求$L(G) = 1 \cdot \log\left(D\left(G(z)\right)\right)$尽可能小。合在一起，便有下面minimax问题：

  For the generator, we expect that the generated images are identical to the real images. Alternatively speaking, the chance of being categorized into real image class is the greater the better. Hence for the loss function $L(G) = 1 \cdot \log\left(D\left(G(z)\right)\right)$, G should maximize its value. Since an alternative form $1 \cdot \log\left(1 - D\left(G(z)\right)\right)$ has appeared previously, we can henceforth minimize this reciprocal term. Put it together, we have the following minimax game problem:

$$\min_G \max_D L(D,G) = \sum_{i=1}^{m} 1 \cdot \log D\left(x_{real}^{(i)}\right) + \sum_{j=1}^{n} 1 \cdot \log\left(1 - D\left(G\left(z^{(j)}\right)\right)\right)$$

对抗式生成网络
Generative Adversarial Networks

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.

自编码器
Autoencoder

- 对于GAN来说，当训练完毕，生成器会生成接近真实样本的样本。一般，这些样本能为我们感知（知道是什么东西）。而还有利用神经网络的另一种模型，称为自编码器，当其训练完毕时，生成的东西可能不能为我们直接理解，我们称为隐表征。这个隐字，是从概率统计里面的隐变量衍生过来的，隐变量指本身不能被直接观测的变量。因此，可以说隐表示是本身不能被直接理解的表示。

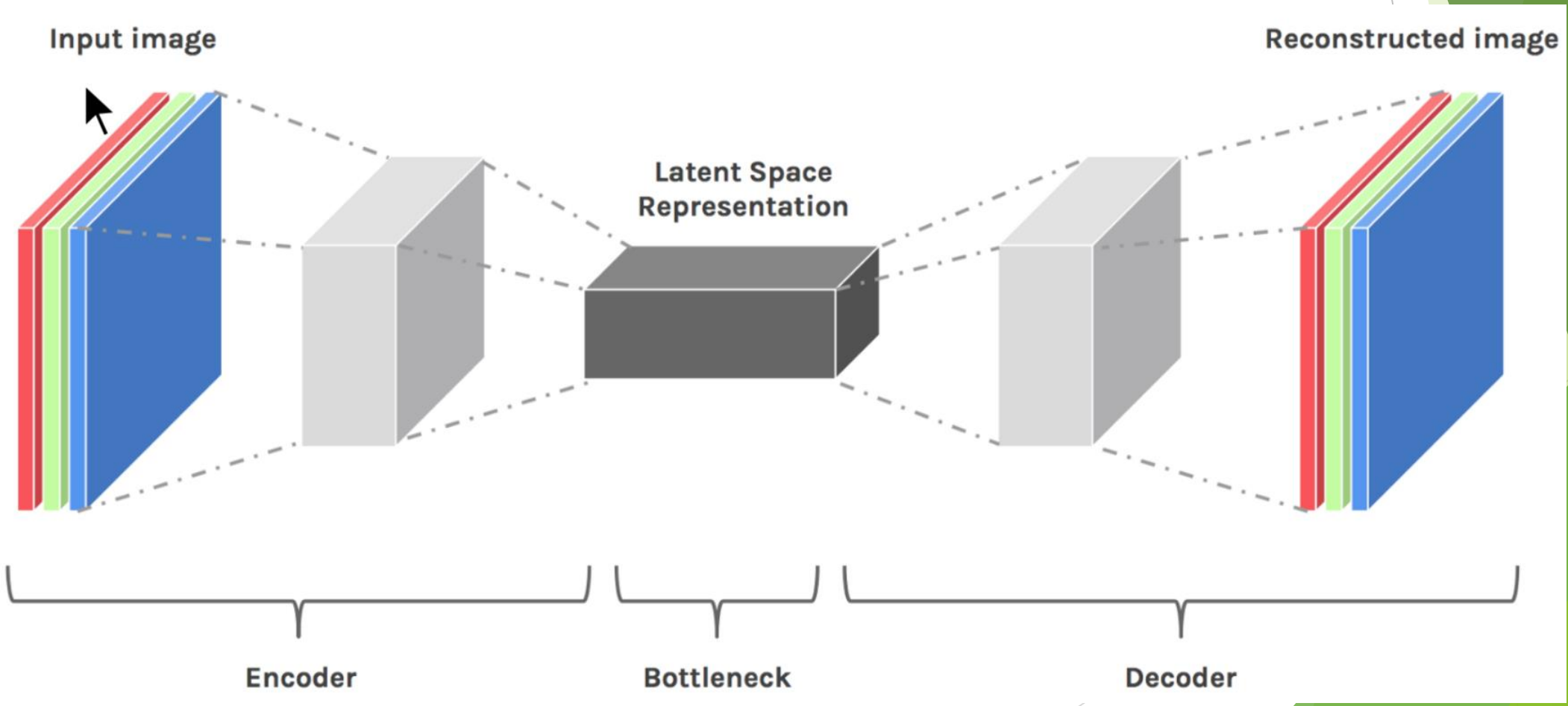
  For GAN, upon finish of training, the generator will output samples resemble the real ones. Generally, these samples are conceivable (we understand these samples). But there also exists another generative model which utilizes neural network, called autoencoder. When training finishes, usually we cannot understand the output content, and we call it latent representation. The wording surrounding “latent” derives from the latent variables from probabilities and statistics. Latent variables themselves cannot be directly observed, which means the latent representations cannot be directly comprehended.

自编码器
Autoencoder

- 为什么需要隐表示呢？比如，对一个图片来说，各部分可能是高度相关的。在某些应用中，可能希望这个图片有另外一种维数较小的紧表示，此即此种模型的初衷。寻求隐表示的过程，我们称为编码，由于在一定程度上，表示要能再现原本的状态，所以要经过解码，与原来的数据进行比较，来优化编码表示。

  The motivation for constructing an auto-encoder can be for dimension reduction. Take an image for example, there might exist highly dependent regions among the picture, and some application might desire a low-dimensional compact representation. The process to find such a representation is called encoding. Because it is also required the latent representation can be converted back to the original data in some fedality, an decoding process is also incorporated to have the reconstructed images to match up with the original ones, so as to optimize the encoder.

自编码器
Autoencoder



习题
Problems

1. 阅读如下链接的材料，尝试实作生成手写数字的GAN的应用。

   Read the following material to implement a generative adversarial network to generate hand-written digits.

   https://www.oreilly.com/content/generative-adversarial-networks-for-beginners/

2. 阅读如下链接的材料，尝试实作生成手写数字的隐表示。

   Read the following material to implement an auto-encoder to generate the latent representation of hand-written digits.

   https://www.edureka.co/blog/autoencoders-tutorial/