

法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920



公众号



微信

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



deepshare.net

深度之眼

Deep Learning

实际工作中的一些高级技术

导师: Johnson

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

主要内容



1. 训练加速
2. 推理加速
3. 深度学习自适应
4. 对抗神经网络

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

训练加速

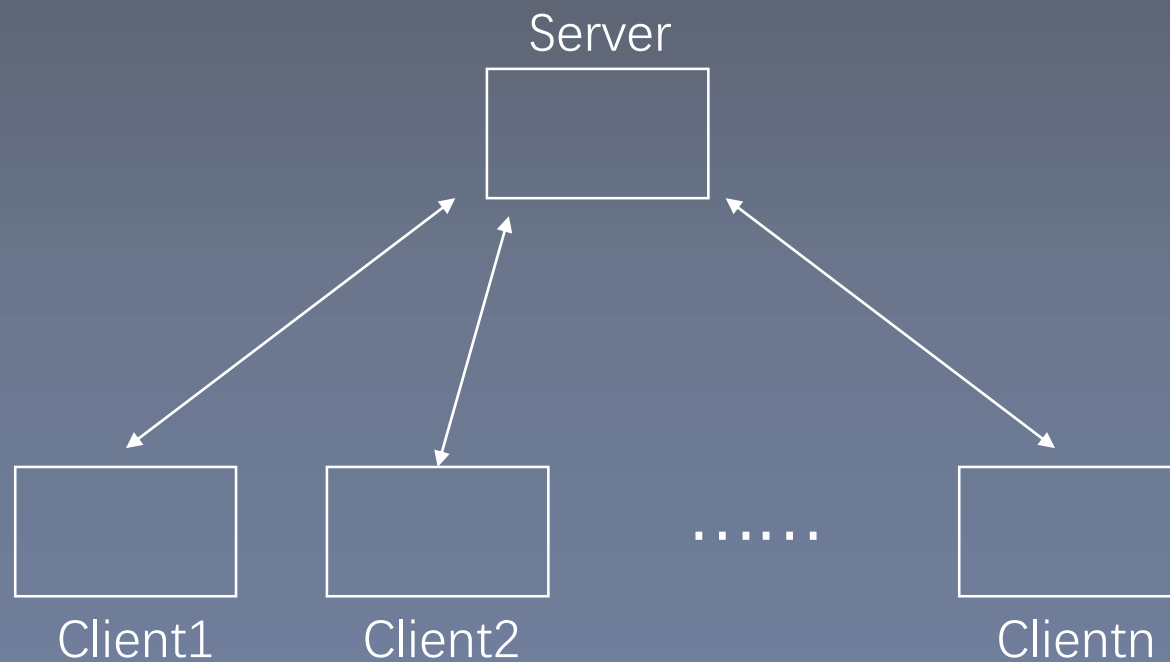


工作中最常用为多GPU训练:

- 1.基于数据的并行（常用）
- 2.基于模型的并行

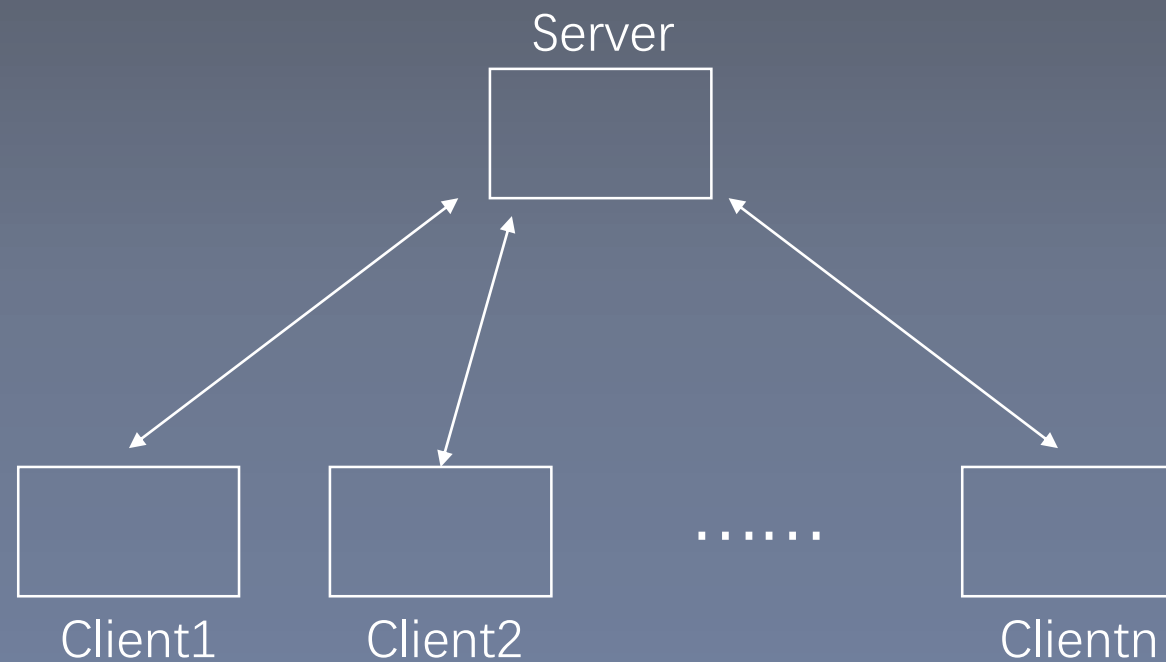
基于数据的并行

Model Average(模型平均)



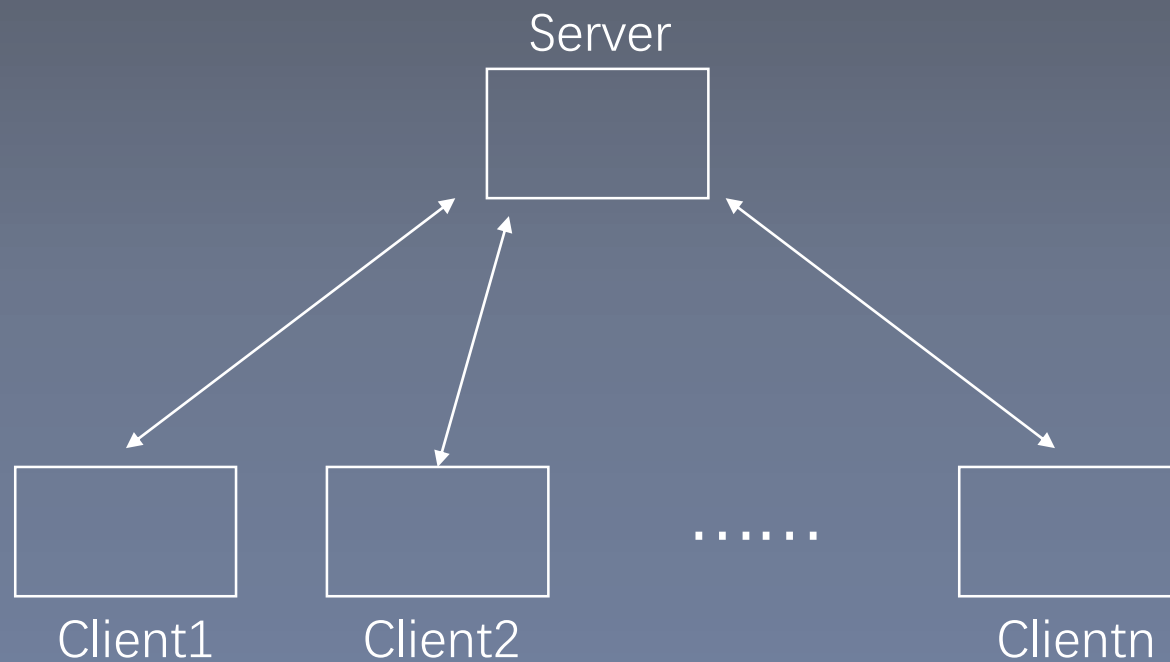
基于数据的并行

SSGD(同步随机梯度下降)



基于数据的并行

ASGD(异步随机梯度下降)



基于模型的并行



关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

推理加速



- 1.SVD分解
- 2.Hidden Node prune
- 3.知识蒸馏(teacher student)
- 4.参数共享(lstm的参数共享)
- 5.神经网络的量化
- 6.Binary Net
- 7.基于fft的循环矩阵加速

SVD分解

1. Pre train
2. svd+fine tuning

$$A_{m \times n} = V^T_{m \times m} \begin{pmatrix} \lambda_1^{\frac{1}{2}} & & \\ & \lambda_2^{\frac{1}{2}} & \\ & & \ddots \end{pmatrix} U_{n \times n}$$

Hidden Node prune



1. Pre train
2. Hidden Node prune+fine tuning

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

知识蒸馏(teacher student)

KL距离 (相对熵)

KL距离，是Kullback-Leibler差异 (Kullback-Leibler Divergence) 的简称，也叫作相对熵 (Relative Entropy)。它衡量的是相同事件空间里的两个概率分布的差异情况。其物理意义是：在相同事件空间里，概率分布 $P(x)$ 对应的每个事件，若用概率分布 $Q(x)$ 编码时，平均每个基本事件（符号）编码长度增加了多少比特。我们用 $D(P || Q)$ 表示KL距离，计算公式如下：

$$D(P || Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

当两个概率分布完全相同时，即 $P(X) = Q(X)$ ，其相对熵为0。

我们知道，概率分布 $P(X)$ 的信息熵为：

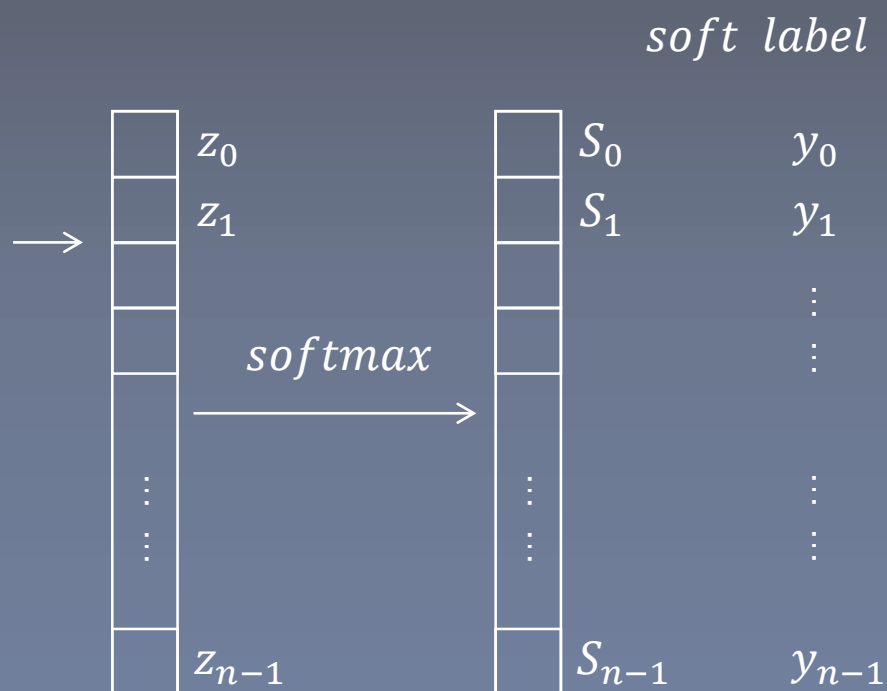
知识蒸馏(teacher student)



deepshare.net

深度之眼

$$D(P \parallel Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$



$$L = - \sum_{i=0}^{n-1} y_i \log S_i$$

$$= - \sum_{i=0}^{n-1} y_i \log \frac{e^{z_i}}{\sum_{j=0}^{n-1} e^{z_j}}$$

$$= - \sum_{i=0}^{n-1} y_i z_i + \sum_{i=0}^{n-1} y_i \log \sum_{j=0}^{n-1} e^{z_j}$$

$$\frac{\partial L}{\partial z_k} = -y_k + \sum_{i=0}^{n-1} y_i \frac{e^{z_k}}{\sum_{j=0}^{n-1} e^{z_j}}$$

$$= -y_k + \sum_{i=0}^{n-1} y_i S_k$$

$$= -y_k + S_k \left(\sum_{i=0}^{n-1} y_i \right)$$

$$= S_k - y_k$$

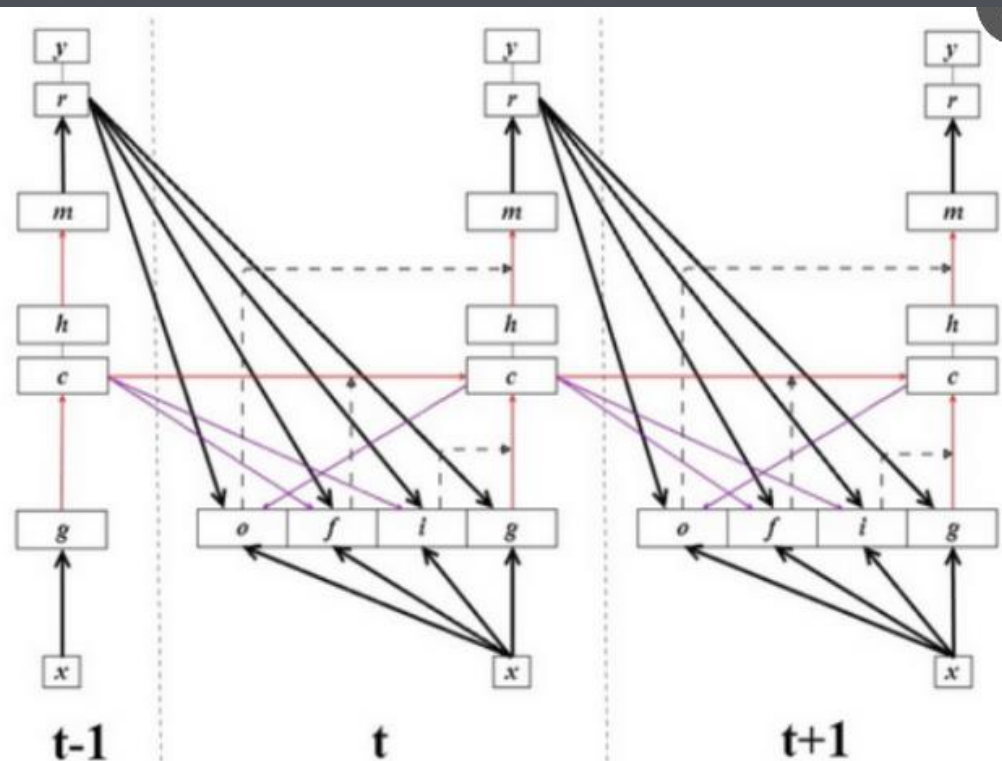
关注公众号深度之眼，后台回复电子书，获取AI必学书籍及完整学习资料

参数共享



deepshare.net

深度之眼



For traditional LSTM:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (4)$$

$$m_t = o_t \odot h(c_t) \quad (5)$$

$$y_t = \phi(W_{ym}m_t + b_y) \quad (6)$$

For recurrent projected layer, substitute (6) with (7) and (8):

$$r_t = W_{rm}m_t$$

$$y_t = \phi(W_{yr}r_t + b_y) \quad (8)$$

(7) 关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

神经网络的量化



deepshare.net

深度之眼

把浮点运算转成正数运算，如32bit,16bit,8bit

运用intel的sse扩张指令集或者arm 的neon进行加速运算(硬件直接运算)

一次能同时运算多条乘法

训练时候前向后向传播算法的微调

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

Binary Net (二值化网络)



参数只有-1,1两个参数

采用位运算进行加速

训练时候前向后向传播算法的微调

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

基于fft的循环矩阵CNN

$$(a_1, a_2, a_3 \cdots \cdots a_n) = C$$

$$\rightarrow \begin{pmatrix} a_1, & a_2, & a_3 \cdots \cdots a_n \\ a_n, & a_1, & a_2 \cdots \cdots a_{n-1} \\ a_{n-1}, & a_n, & a_1 \cdots \cdots a_{n-2} \\ \vdots & \vdots & \vdots \\ a_2 \cdots \cdots & & a_1 \end{pmatrix} = W$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad Wx = \begin{pmatrix} a_1 x_1 + \cdots a_n x_n \\ a_n x_1 + \cdots a_{n-1} x_n \\ \vdots \\ \vdots \end{pmatrix} = C * x$$

$$FF_t(c * x) = FF_t(c) \cdot FF_t(x)$$

$$c * x = FF_t^{-1}[FF_t(c) \cdot FF_t(x)]$$

深度学习自适应(增量学习)



1. 初始参数的网络迁移
2. 场景自适应(kld)
3. 数据的混合

初始参数的网络迁移



用已有数据集如image net先训练一个模型，在新数据上以此模型
作为初始模型做fine tuning

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

场景自适应(kld)



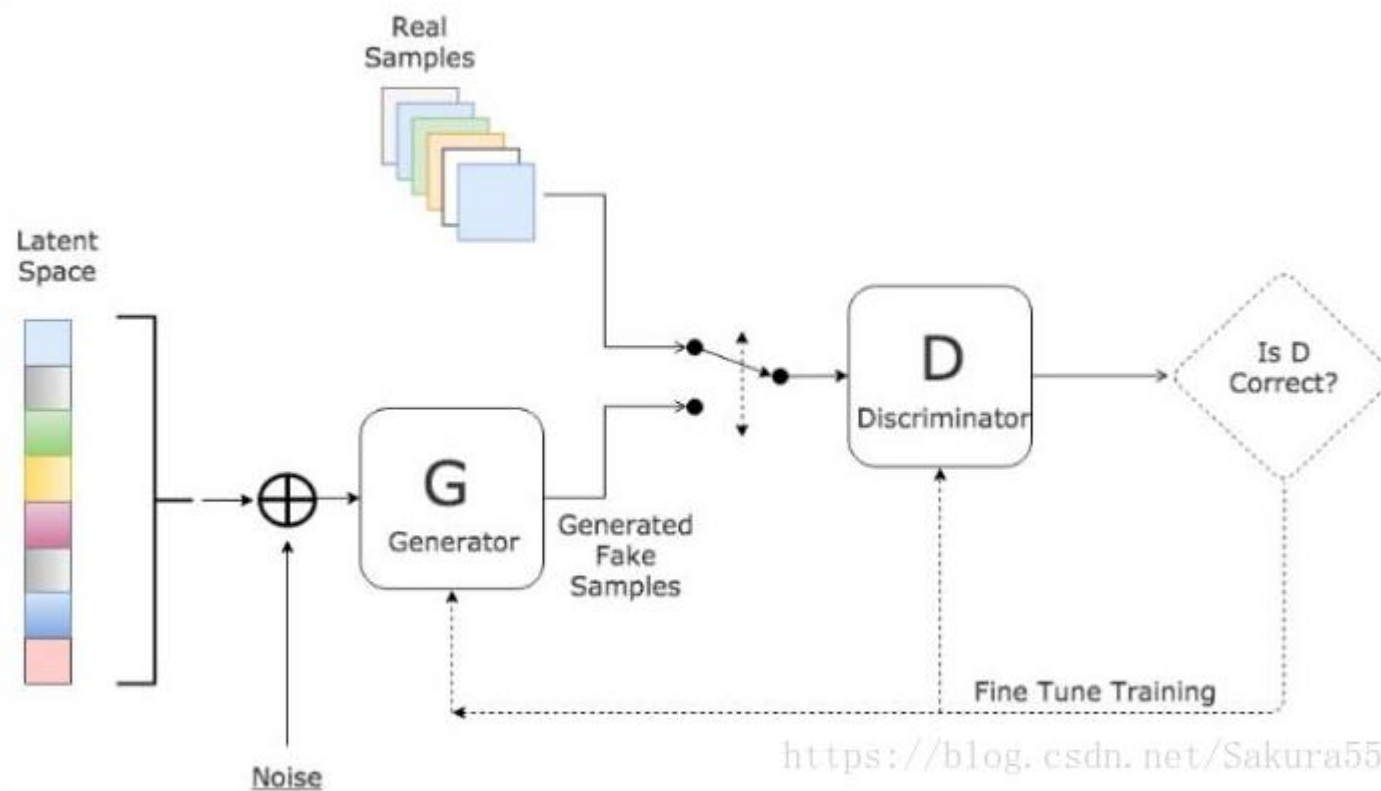
$$J = \alpha \cdot J_1(x, y, w) + (1 - \alpha) \cdot J_2(x, \tilde{w}, w)$$

数据的混合（效果好，训练慢，难以快速迭代）

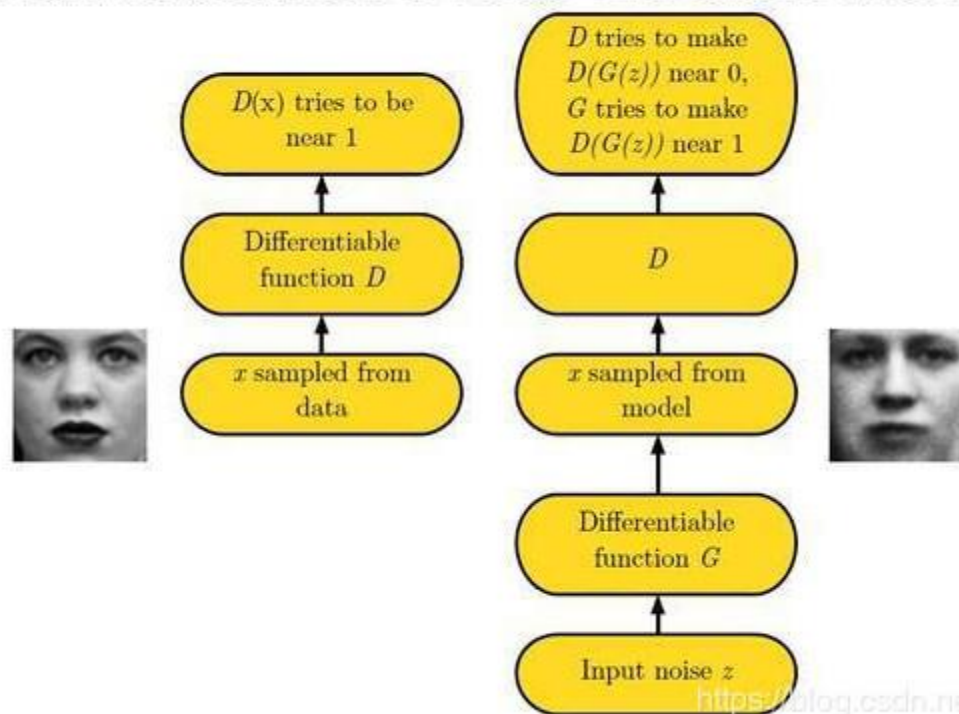
1. 新增数据量比较大
2. 新增数据量比较小

对抗神经网络

对抗生成网络 (Generative Adversarial Net)



Adversarial Nets Framework



deepshare.net

深度之眼

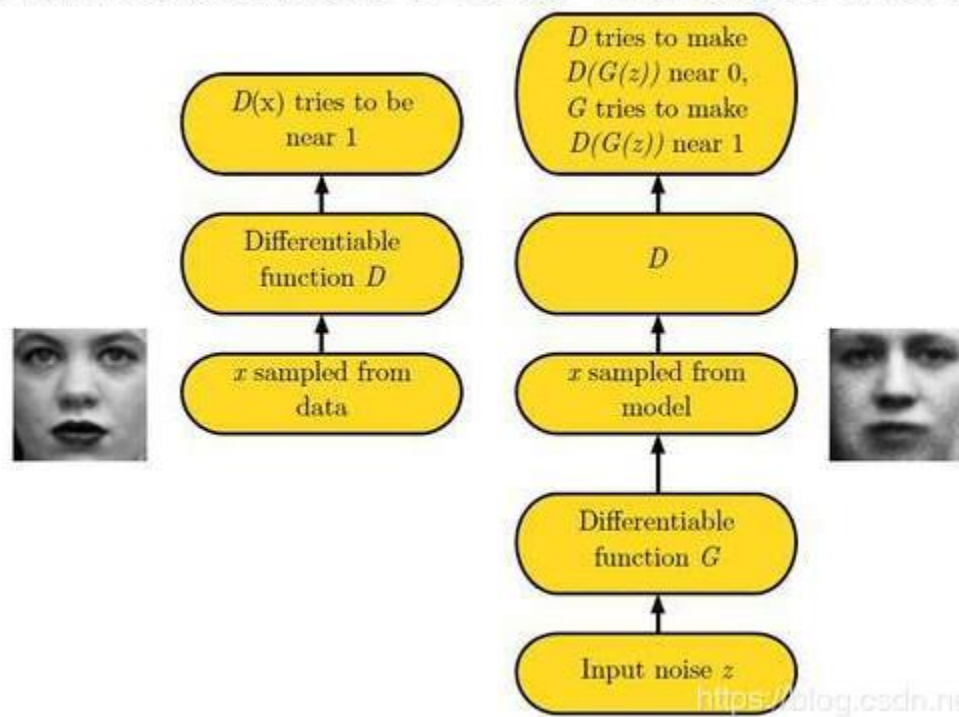
GAN的基本原理其实非常简单，这里以生成图片为例进行说明。假设我们有两个网络，G (Generator) 和D (Discriminator)。正如它的名字所暗示的那样，它们的功能分别是：

- 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”。

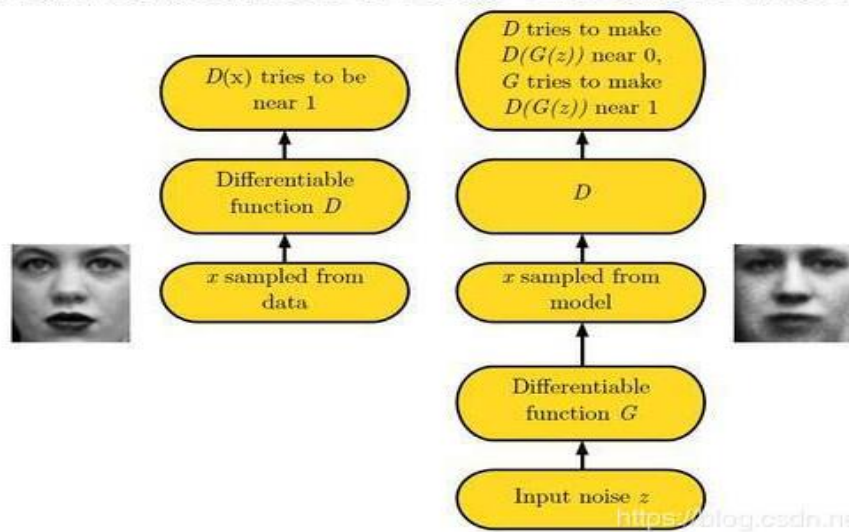
关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

Adversarial Nets Framework



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

Adversarial Nets Framework



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

for number of training iterations **do**

for k steps **do**

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

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

end for

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

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

end for

——结 语——

恭喜同学们顺利完成本课程的全部内容
请大家在课下一定要好好总结、回顾和应用知识点
祝大家在AI学习之路上越走越远



关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



deepshare.net

深度之眼

联系我们：

电话：18001992849

邮箱：service@deepshare.net

QQ：2677693114



公众号



客服微信

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料