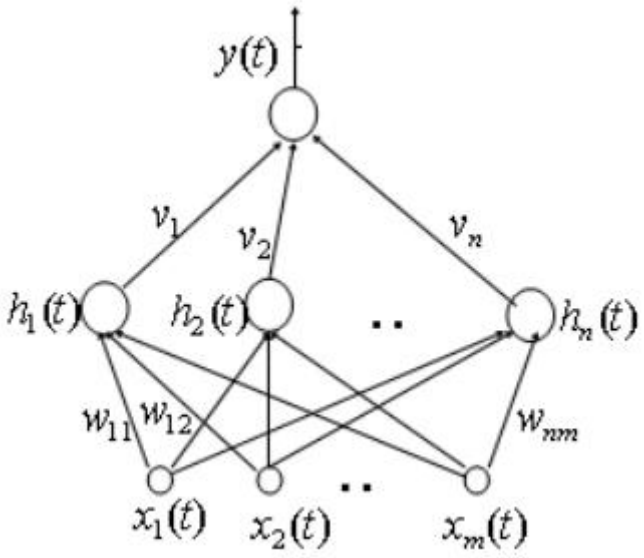
Deep Learning: An Introduction

说明

- 入门级的介绍，为了便于理解尽量介绍insight清晰的内容，而不是后续改进算法。
- 近几年新的东西很多（也许过多），新进展难言全面了解

Beyond 3 layers

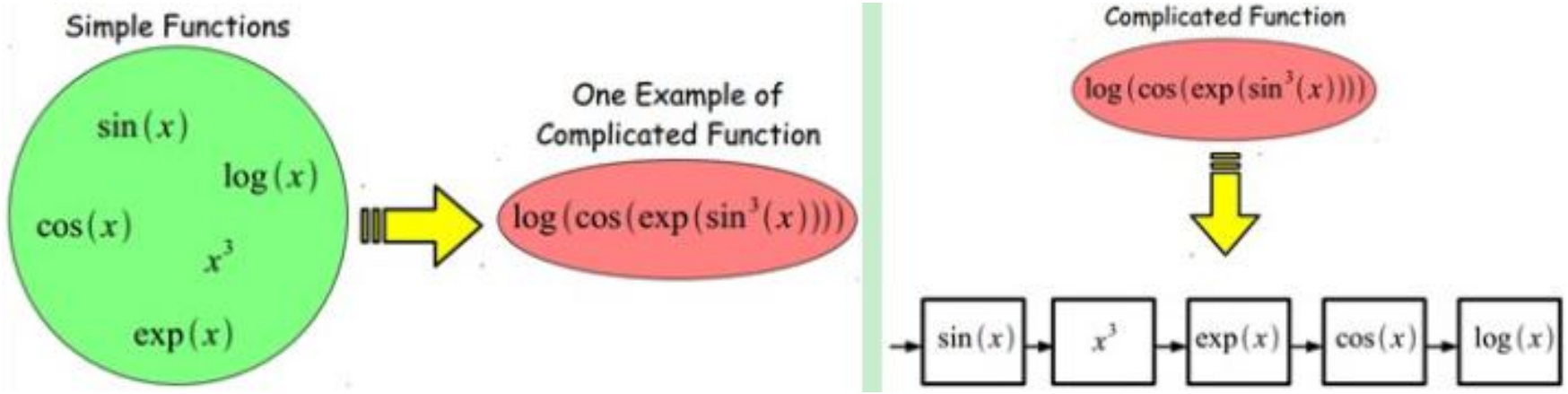
Cybenko's Theorem



Why more than 3 layers?

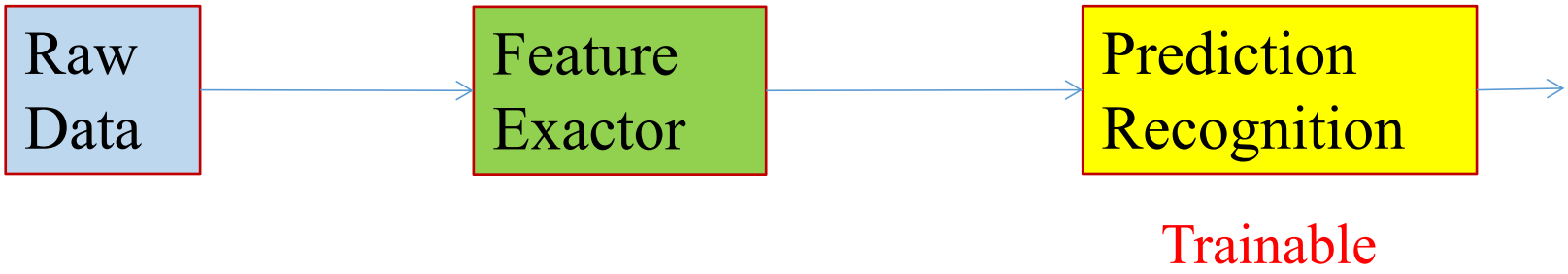
从复杂函数逼近的角度

- 有限样本和计算单元情况下，3层网络对复杂函数的表示能力有限
- 嵌套比加权和更有效
- 知道逼近哪个函数？



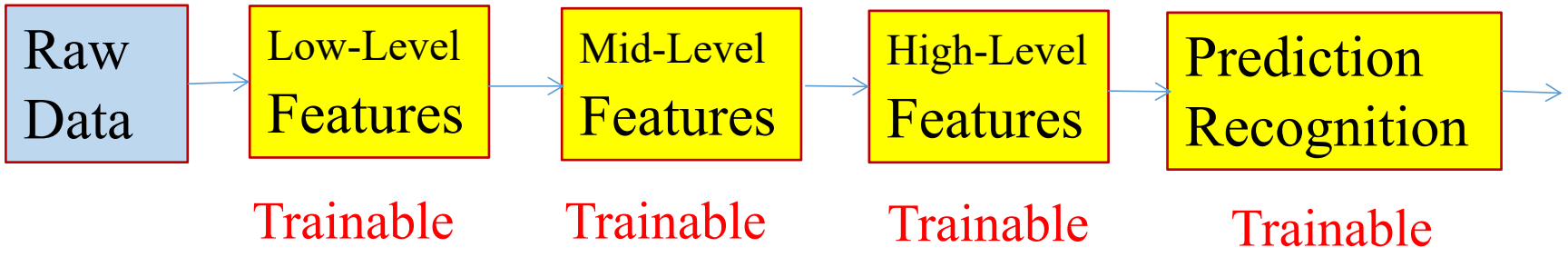
从特征表示的角度

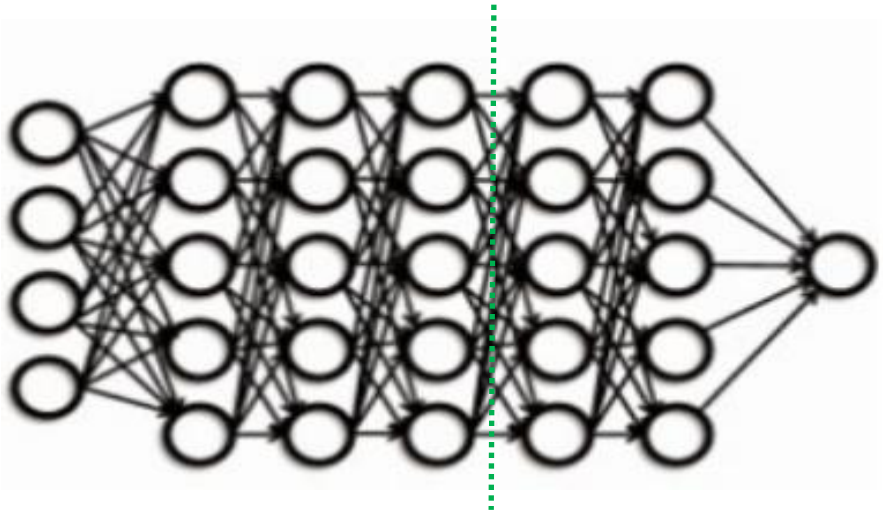
➢**传统的学习/模式识别**



- 好的特征是识别成功的关键，特征更重要
- 传统机器学习使用固定/人工设计的特征
- 不同研究对象特征不同，特征具有多样性，手工选取特征费时费力，需要启发式专业知识，很大程度上靠经验和运气

➢深度学习

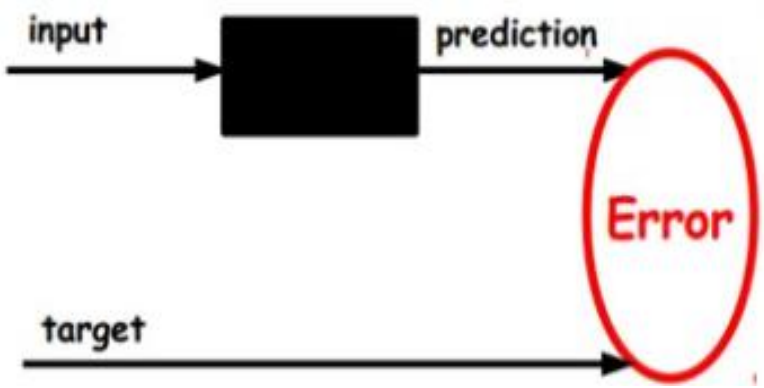


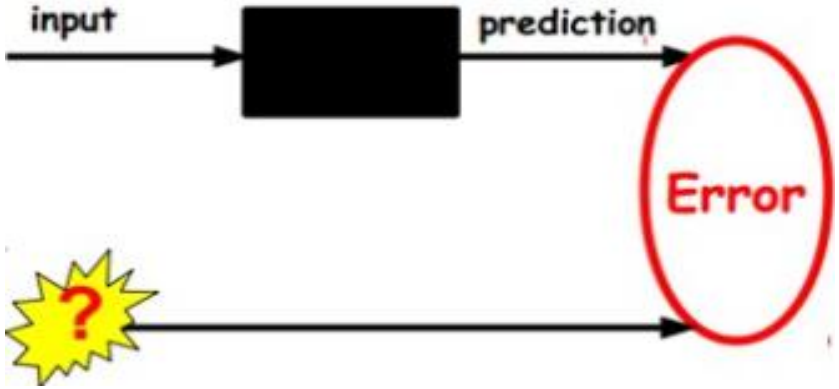


多隐层的人工神经网络具有优异的特征学习能力，学习得到的特征对数据有更本质的刻画，从而有利于预测和识别

训练多层网络

- 训练多层网络：BP算法
- 理论上可逼近 VS 实际能找到合适的权值
- 用BP算法训练多层网络存在的问题：
  - 理论上，由于是采用随机值初始化，当初值是远离最优区域时易收敛至局部极小点。LeCun等说虽然大家都这么认为，实际中这个问题不大，主要是过拟合导致推广能力差。[Nature 2015]
  - 梯度消散，到前些层时，$<10^{-10}$量级
  - BP算法需要有大量有标签数据来训练，但获得大量数据标签通常代价昂贵/不可能
- 复杂模型训练得不好，效果未必比简单模型好。





深度学习

- 深度（多层）神经网络在训练上的难度，可以通过“逐层初始化”（layer-wise pre-training）来有效克服，逐层初始化可通过无监督学习实现
- 深度学习：一种基于无监督特征学习和特征层次结构的学习方法（未必都认同）
- 可能的的名称：深度学习、表征学习、无监督特征学习等

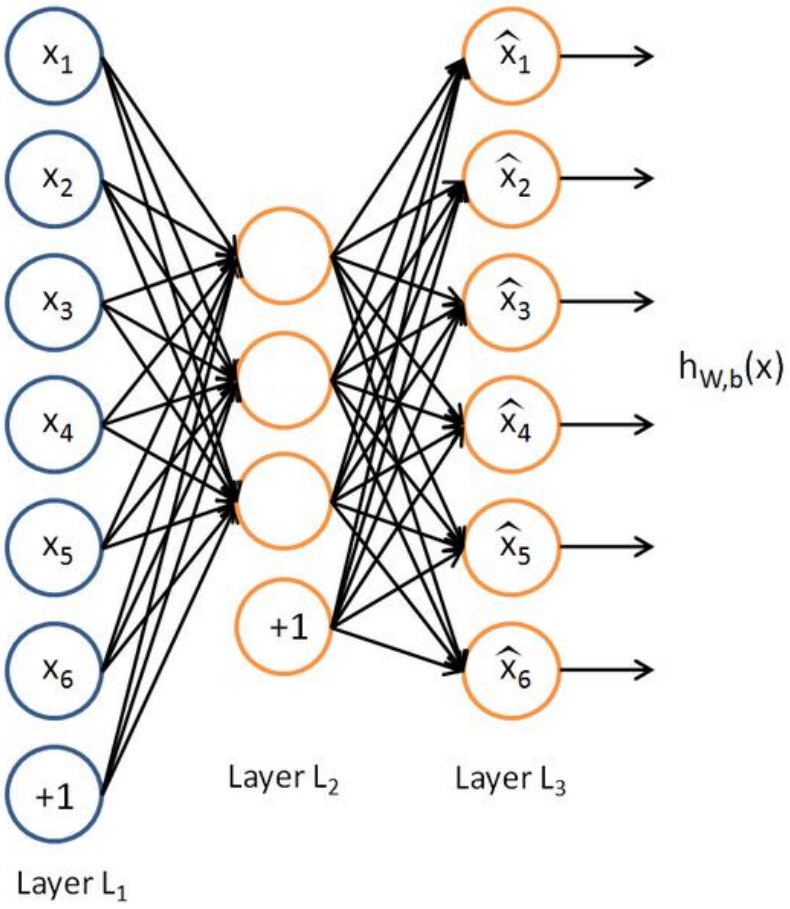
代表性模型：

- Autoencoder
- Deep Belief Network / Restricted Boltzmann Machine
- Convolutional Neural Network
- Generative Adversarial Nets (GAN)
- Long Short Term Memory

Autoencoder

Autoencoder

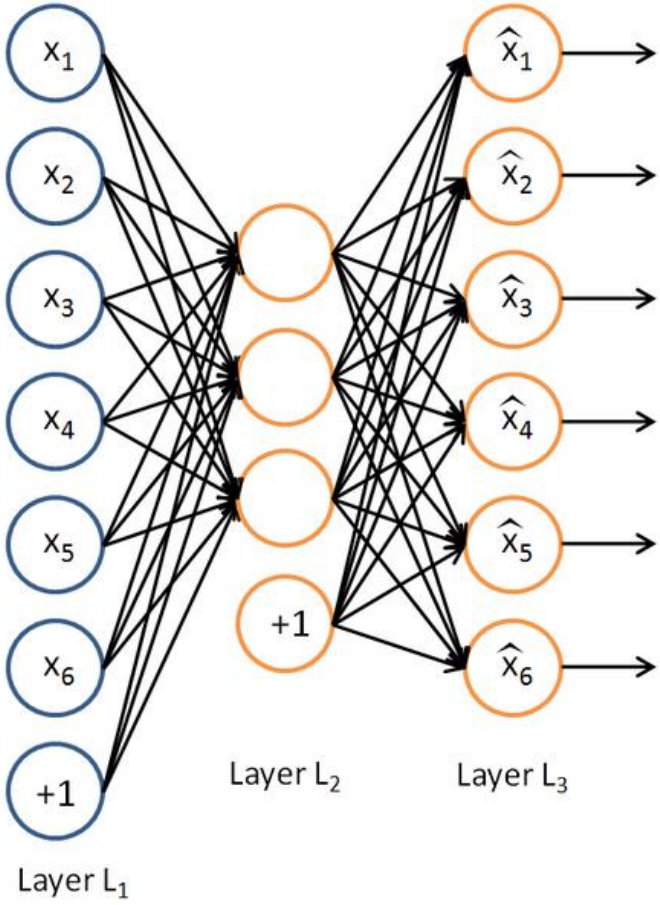
aka: Autoassociator ~1988

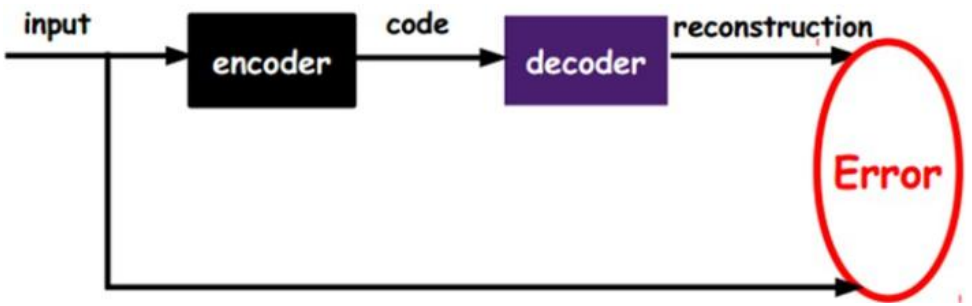


- 无监督学习算法
- 只有一层隐层节点，输入和输出具有相同节点数的神经网络
- 自动编码器训练的目标是

$$h_{w,b}(x) \approx x$$

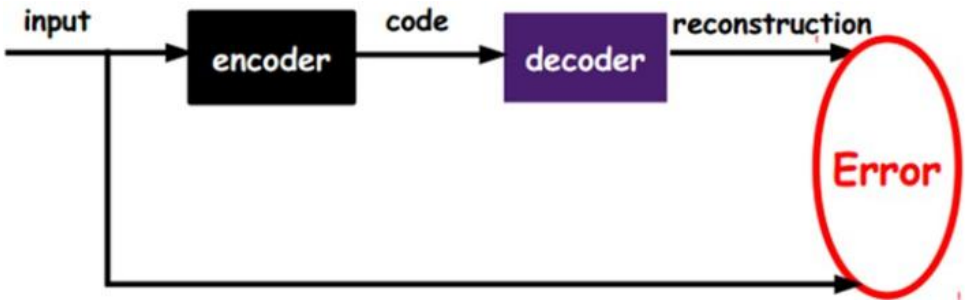
- 可以用BP训练
- 用来干啥？ 特征提取
- 线性激活函数的情况下，等价于PCA

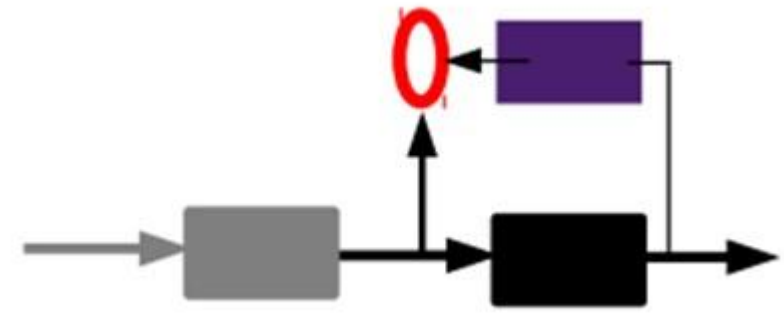


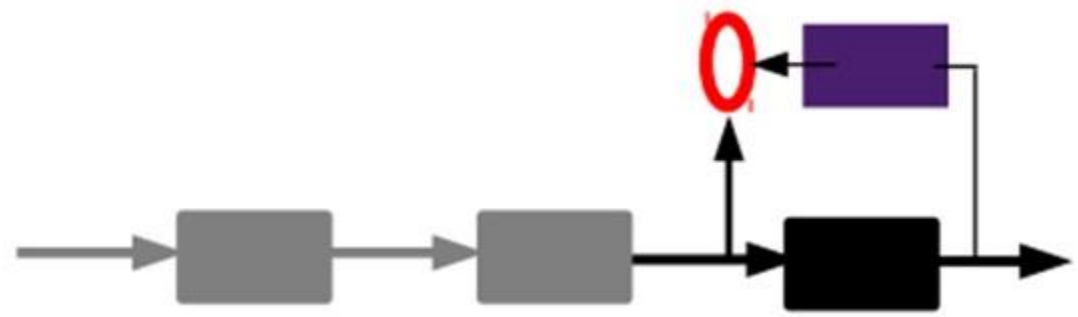


我们通过调整encoder和decoder的参数，使得重构误差最小，这时候我们就得到了输入input信号的第一个表示了，也就是编码成code了。因为是无标签数据，所以误差的来源就是直接重构后与原输入相比得到。

逐层预训练：Stacked Autoencoder







- 得到了第一层的code，重构误差最小让我们相信这个code就是原输入信号的良好表达了
- 第二层和第一层的训练方式没有差别，将第一层输出的code当成第二层的输入信号，同样最小化重构误差，就会得到第二层的参数，并且得到第二层的code，也就是原输入信息的第二个表达了
- 逐层用同样的方法训练
- 训练这一层，前面层的参数都是固定的，并且他们的decoder已经没用了，不需要了

有监督微调

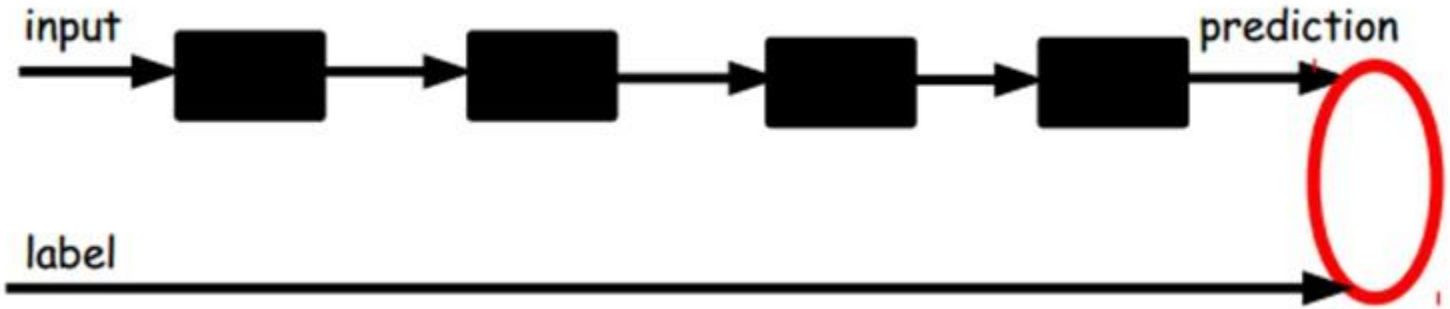
- 经过上面的方法，就有很多层了，每一层得到了原始输入的不同的表达，至于需要多少层，目前没有科学的确定方法，需要自己试调。
- 到现在，这个垒叠的Autoencoder还不能用来分类数据（预测），因为它还没有学习如何去连接一个输入和一个类。它只是学习获得了一个可以良好代表输入的特征，这个特征可以最大程度上代表原输入信号。
- 为了实现分类（预测），我们可以在stacked Autoencoder最后的编码层后添加一个分类器（例如logistic回归、SVM等）
- 然后通过监督训练方法去训练

有监督微调(fine-tuning)

通过监督学习进行微调，也分两种：

一个是只调整分类器/预测器（从特征提取的角度看预训练）

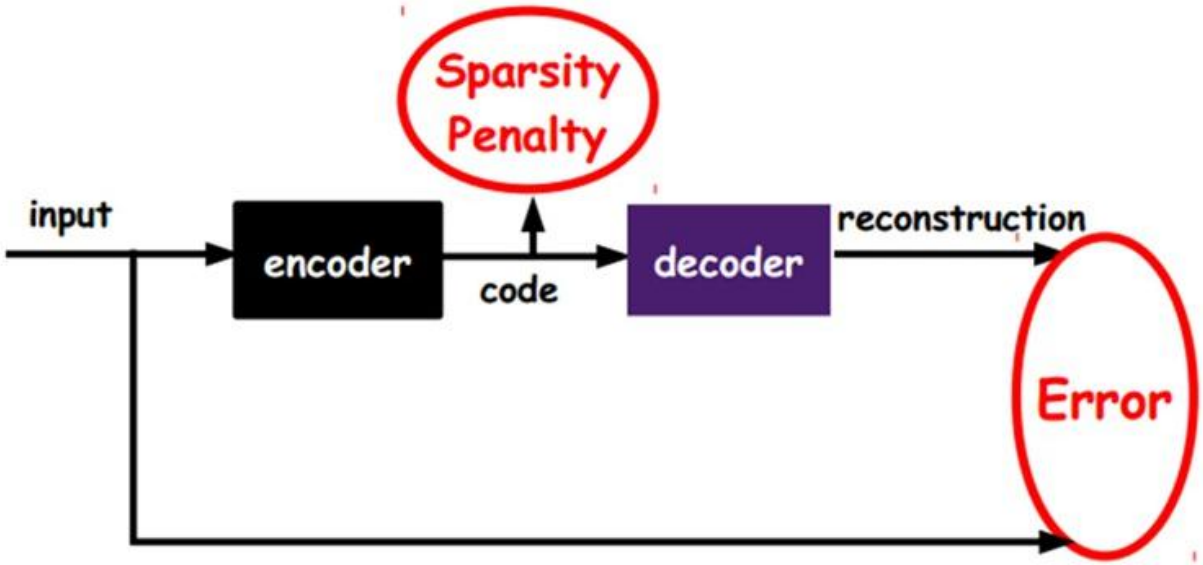
另一种：通过有标签样本，微调整个系统（从函数逼近初始化的角度看预训练）



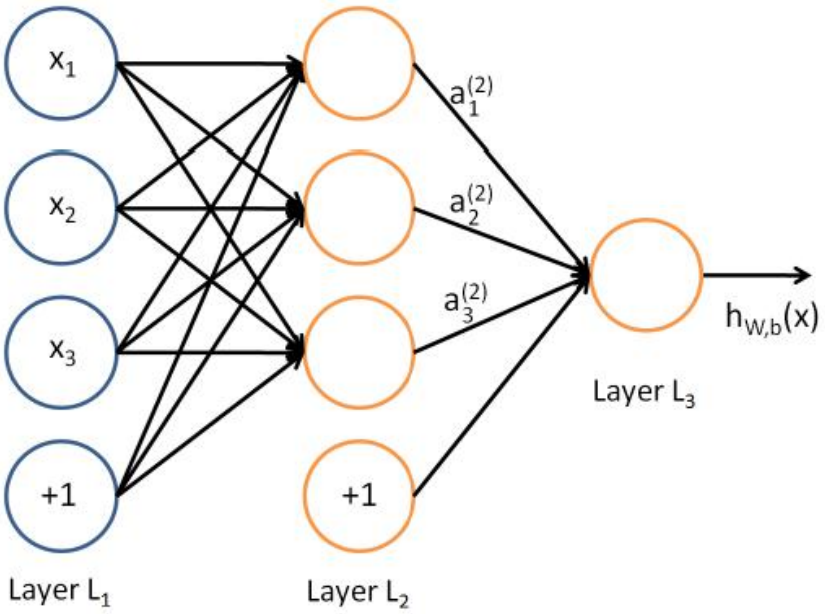
如果有足够多的有标签数据，后者更好

Sparse Autoencoder

- 如果隐层节点比输入、输出少的话，由于被迫降维，自编码器会自动习得训练样本的特征（变化最大，信息量最多的维度）。但是如果隐层节点数目过多，甚至比可视节点数目还多的时候，自编码器会丧失这种能力。
- 我们对隐层节点进行稀疏性限制。当隐层神经元的输出接近于1的时候我们认为它被激活，而输出接近于0的时候认为它被抑制，那么使神经元大部分时间都是被抑制的限制则被称作稀疏性限制。



Sparse Autoencoder



$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right]$$

$\hat{\rho}_j$ 为隐藏单元 j 的平均激活值

一般强制约束 $\hat{\rho}_j \to \rho$

ρ 是一个稀疏参数，一般取接近于0的值，比如0.05；

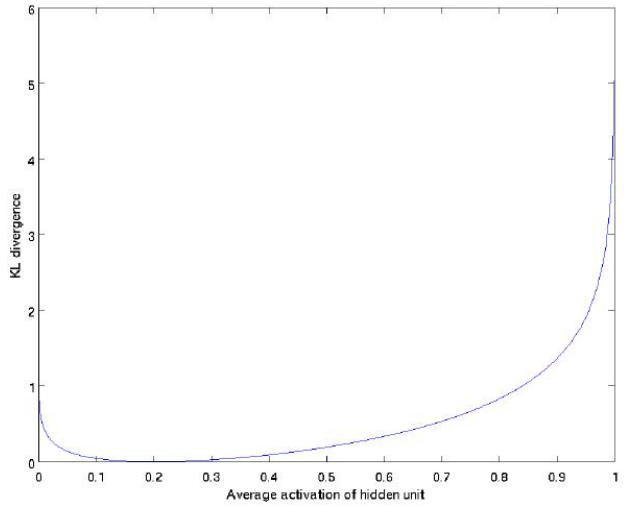
也就是说，每个隐藏单元j的平均激活值接近于0.05。

Sparse Autoencoder

我们添加一个额外的惩罚项至目标函数

$$\mathrm{KL}(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

该式的一个作用是：对 $\hat{\rho}_j$ 偏离 ρ 的程度进行惩罚



左图中，ρ=0.2

当 $\hat{\rho}_j$=0.2 时，KL散度值达到最小，其值为0。

当 $\hat{\rho}_j$ 趋于0或1时，KL散度值趋于无穷大。

因此，为了使以上惩罚项最小，必须使得：

$$\hat{\rho}_j \to \rho$$

惩罚函数的含义：$\hat{\rho}_j$ 可看做0-1分布 [1- $\hat{\rho}_j$ $\hat{\rho}_j$]，KL为与[1-ρ,ρ]的差距

Sparse Autoencoder

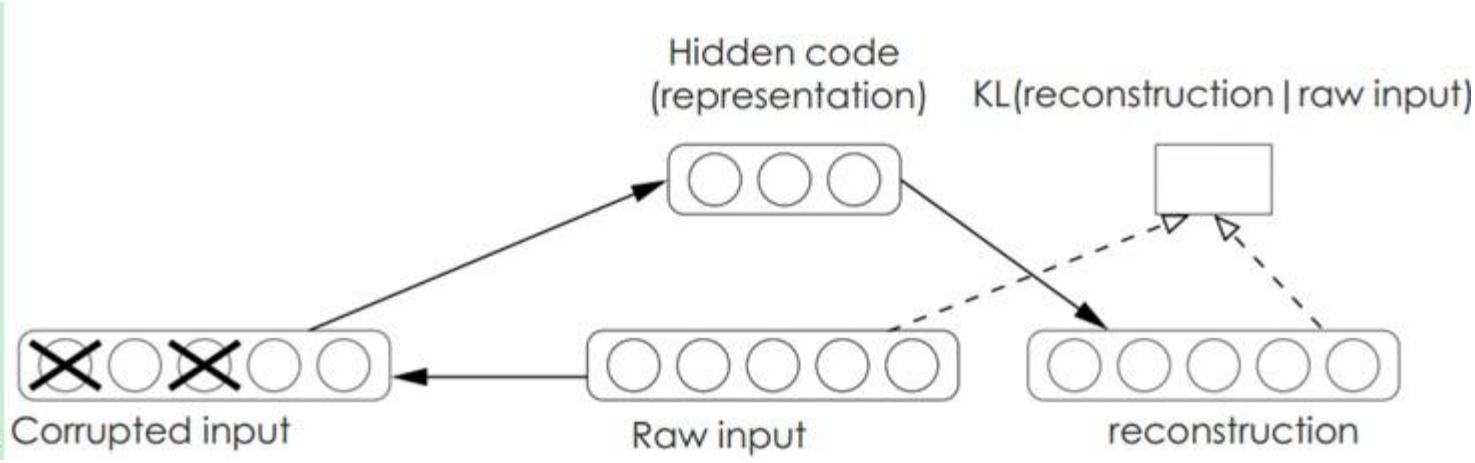
全局损失函数为：

$$J_{\text{sparse}}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} \text{KL}(\rho||\hat{\rho}_j)$$

Stacked Sparse Autoencoder

Denoising Autoencoder

- dAE在Autoencoder的基础上，训练数据加入噪声，所以自编码器必须学习去去除这种噪声而获得真正的没有被噪声污染过的输入。因此，这就迫使编码器去学习输入信号的更加鲁棒的表达，这也是它的泛化能力比一般Autoencoder强的原因。
- 按照原文的意思，Corrupted input 并不是加入高斯噪声，而是以一定概率使输入层节点的值清为0。



Denoising Autoencoder

- Bengio对dAE的直观解释：
- dAE有点类似于人体的感官系统，比如人眼看物体时，如果物体某一小部分被遮住了，人依然能够将其识别出来
- 多模态信息输入人体时（比如声音，图像等），少了其中某些模态的信息有时影响也不大
- 普通的autoencoder的本质是学习一个恒等函数，即输入和重构后的输出相等，这种表示有个缺点就是当测试样本和训练样本不符合同一分布，相差较大时，效果不好。显然，dAE在这方面的处理有所进步

Denoising Autoencoder

Stacked Denoising Autoencoder

- The corruption process is only used during training, but not for propagating representation from the raw input to higher-level representatons.
- When layer k is trained, it receives as input the uncorrupted output of the previous layers
- After layer-wise initialization, in the fine-tuning stage, it is a normal MLP.

几点说明

- 激活函数可用sigmoid函数等
- 输入归一化 [0,1]
- 损失函数 $L(r,x)$，用于测量重建的好坏，目标是最小化L。L除了squared error外，还常使用cross-entropy.

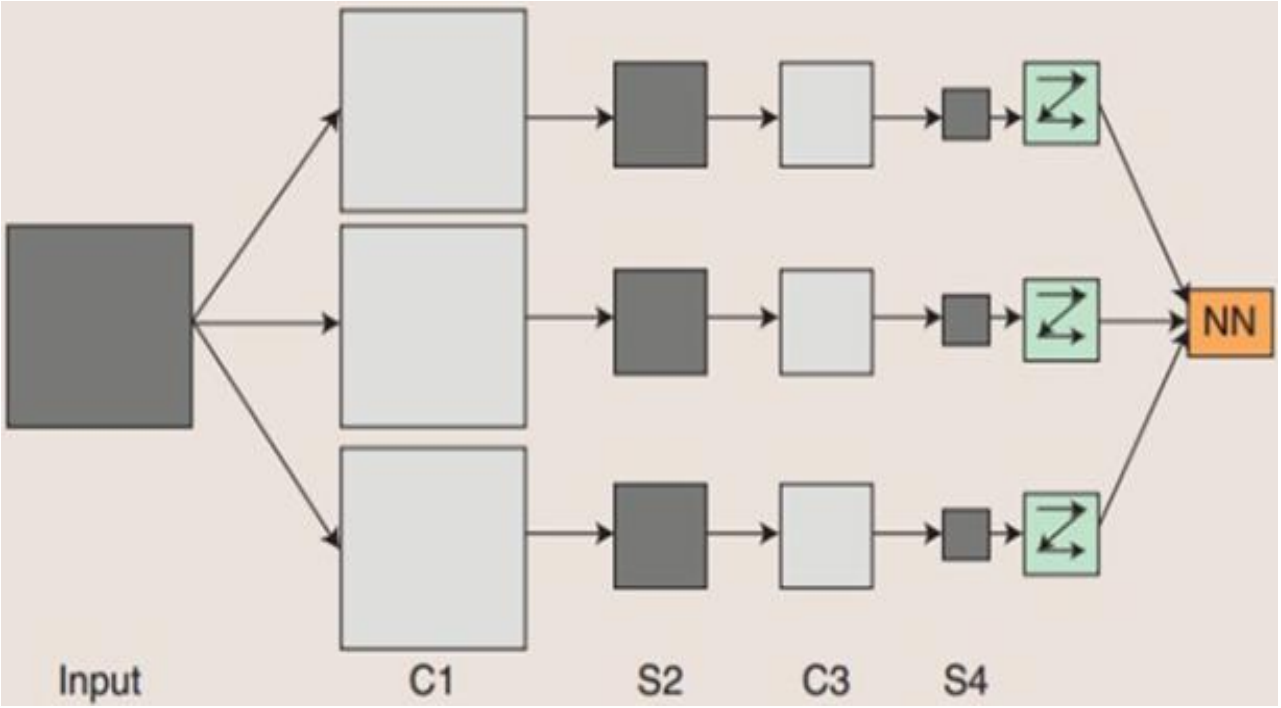
$$L_H(x,z) = -\sum_{k=1}^{d}[x_k \log z_k + (1-x_k)\log(1-z_k)]$$

$$CE(x,z) = H(x) + KL(x,z)$$

- pre-training + fine-tuning 是许多深度学习方法的主要流程

Convolutional Neural Networks

- CNN是为识别二维图像信息而特殊设计的一个多层感知机网络



概念示范：输入图像通过与m个可训练的滤波器和可加偏置进行卷积，在C1层（卷积层）产生m个特征映射图，然后特征映射图中每组的n个像素再进行求和，加权值，加偏置，通过一个Sigmoid函数得到m个S2层（subsampling/pooling）的特征映射图。这些映射图再进过卷积得到C3层。这个层级结构再和S2一样产生S4。最终，这些像素值被光栅化，并连接成一个向量输入到传统的神经网络，得到输出。

CNN概述

CNN的关键技术：

局部感受野、权值共享、下采样

CNN的优点：

1、避免了显式的特征抽取，而隐式地从训练数据中进行学习；

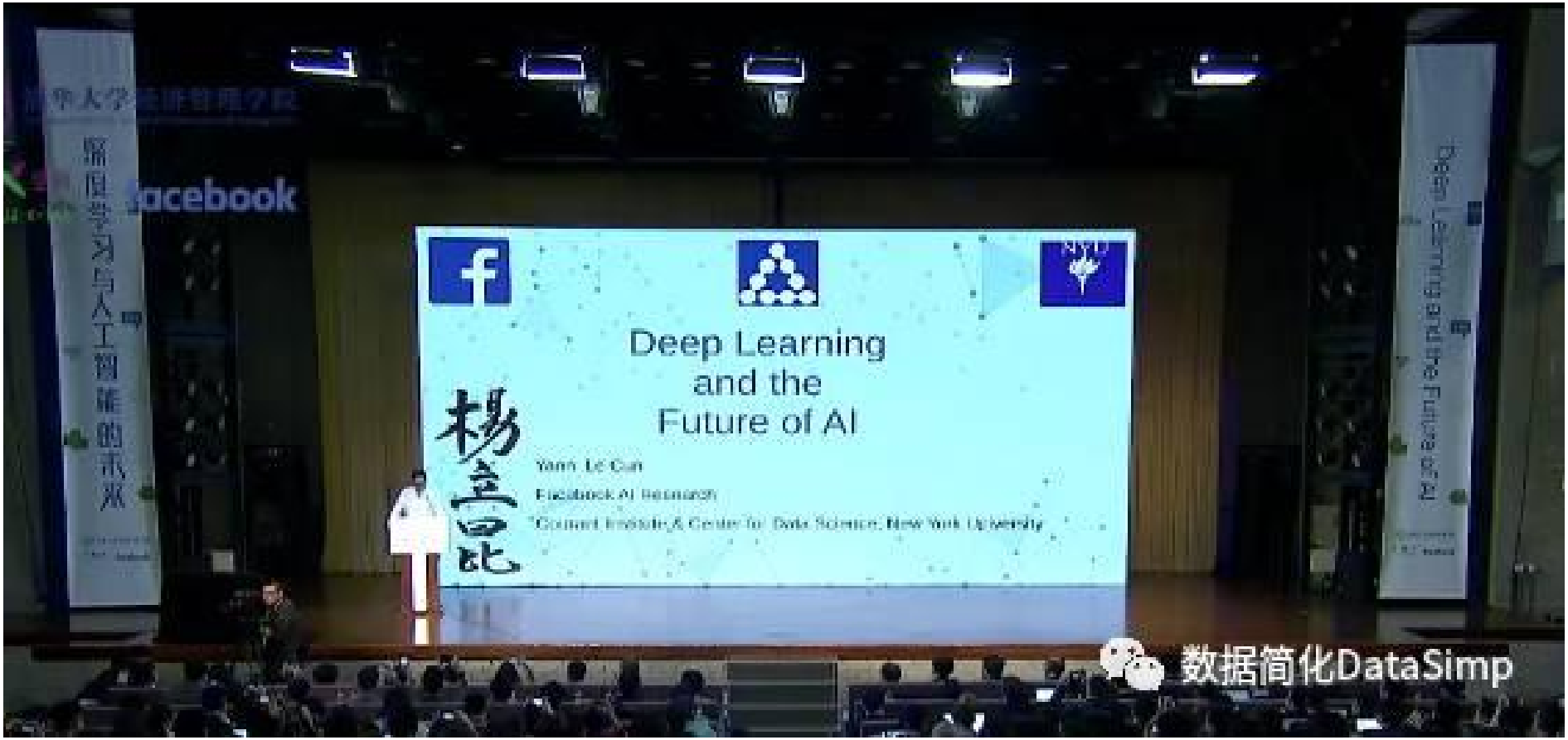
2、同一特征映射面上的神经元权值相同，从而网络可以并行学习，降低了网络的复杂性；

3、采用下采样结构，可以获得一定程度的位移、尺度、形变鲁棒性；

4、输入信息和网络拓扑结构能很好地吻合，在图像处理、语音识别等方面有着独特优势。

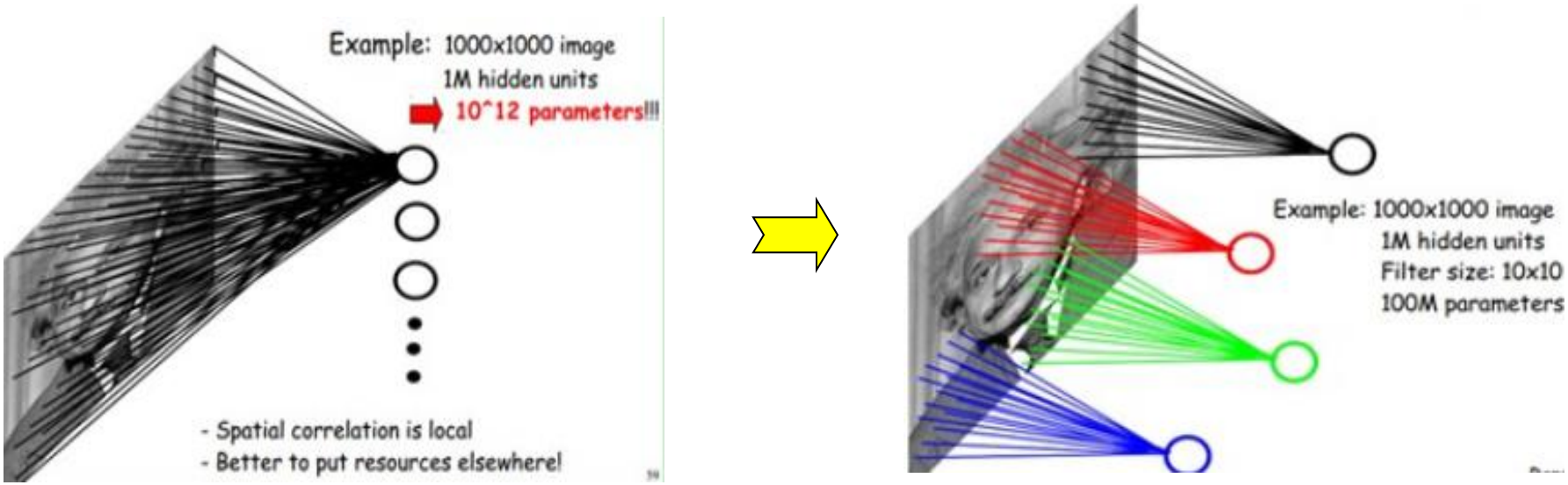
- 20世纪60年代，Hubel和Wiesel提出了卷积神经网络
- 训练、实用化、推广、成为深度学习的代表，主要是Yann LeCun的贡献

Yann LeCun（杨立昆）



注：2017年3月22日下午，Facebook人工智能研究院院长、纽约大学终身教授Yann LeCun在清华大学演讲，在PPT中给出中文名“杨立昆”。

CNN的部件



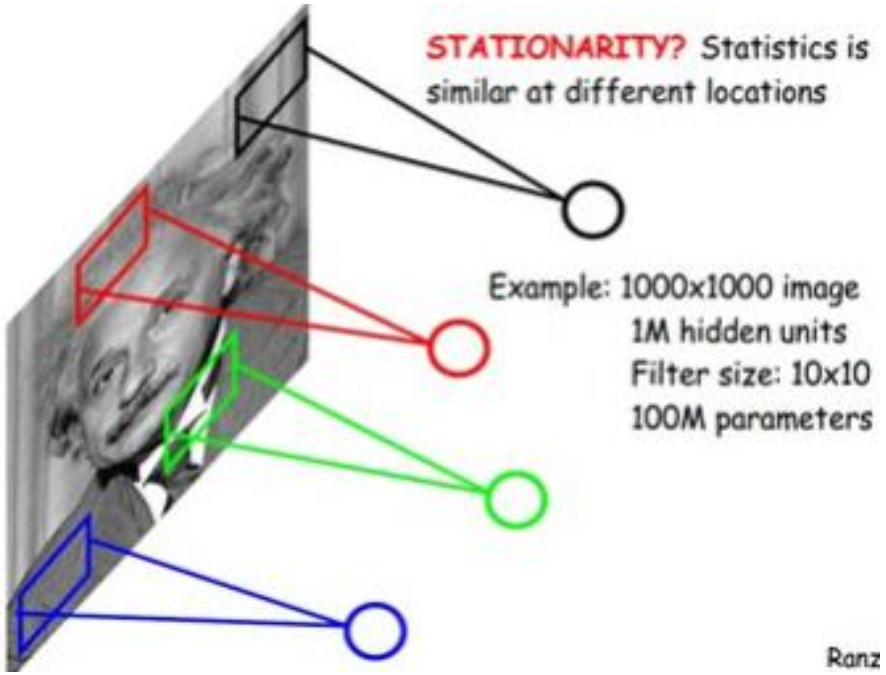
上图左：全连接网络。如果我们有1000x1000像素的图像，有1百万个隐层神经元，每个隐层神经元都连接图像的每一个像素点，就有1000x1000x10^6=10^12个权值参数。

减少参数的方法：

- 神经元感受局部区域(Receptive Field)。上图右为**局部连接**网络，每一个节点与上层节点同位置附近10x10的窗口相连接，则1百万个隐层神经元就只有10^6乘以100，即10^8个参数。
- 每个神经元参数设为相同，即**权值共享**，也即每个神经元用同一个卷积核去卷积图像。不论图像多大，隐层多少神经元，只100个参数。

CNN的部件

- 隐层神经元数量的确定

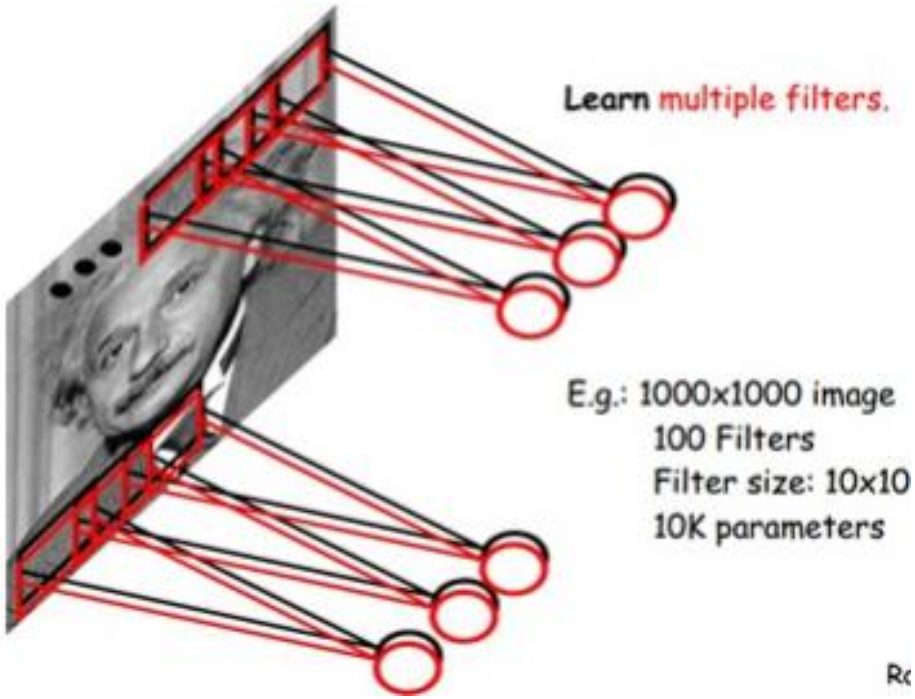


神经元数量与输入图像大小、滤波器大小和滤波器的滑动步长有关。

例如，输入图像是1000x1000像素，滤波器大小是10x10，假设滤波器间没有重叠，即步长为10，这样隐层的神经元个数就是(1000x1000)/(10x10)=10000个。

CNN的部件

- 多滤波器情形



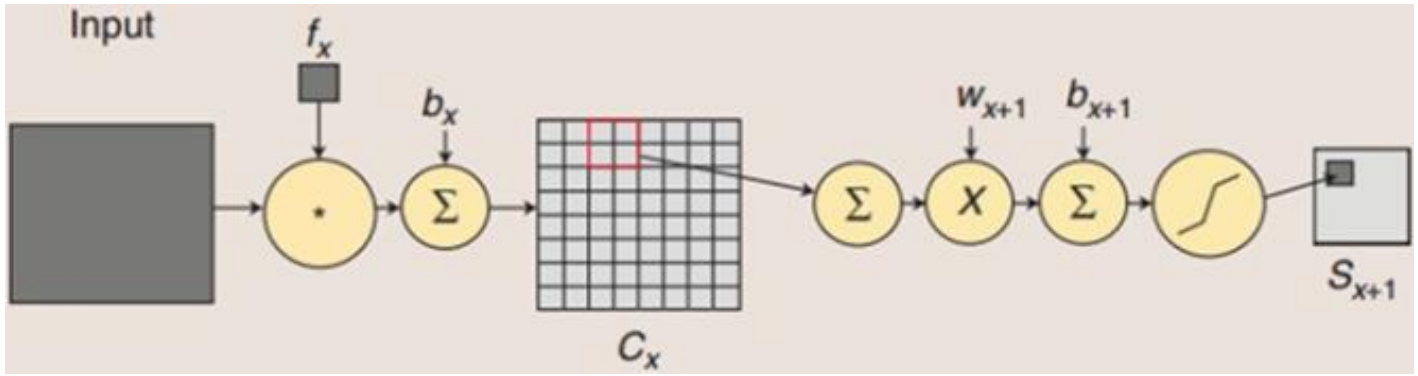
□隐层神经元的个数按滤波器种类的数量翻倍

□隐层参数个数仅与滤波器大小、滤波器种类的多少有关

例如：隐含层的每个神经元都连接10x10像素图像区域，同时有100种卷积核（滤波器）。则参数总个数为：(10x10+1)x100=10100个

CNN的部件

- 计算



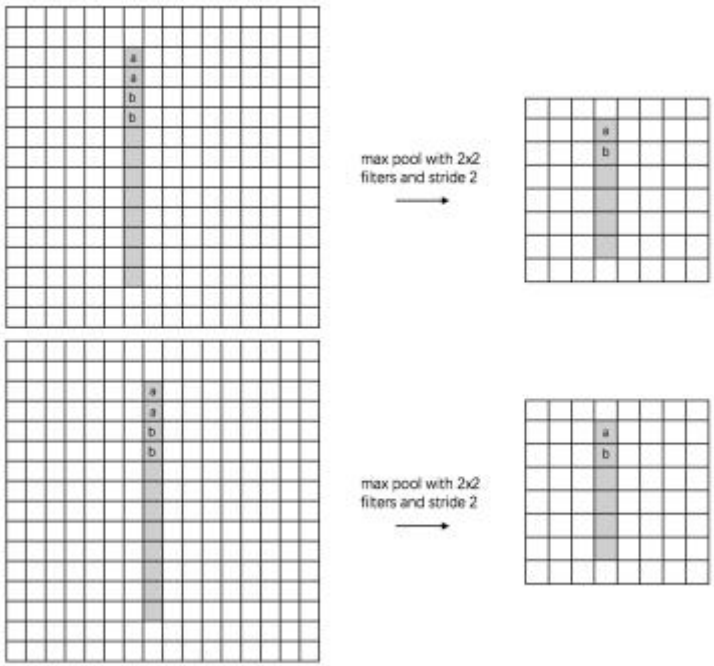
□卷积过程：用一个可训练的滤波器f_x去卷积一个输入的图像（第一阶段是输入的图像，后面的阶段就是S层输出了），然后加一个偏置b_x，得到卷积层C_x。

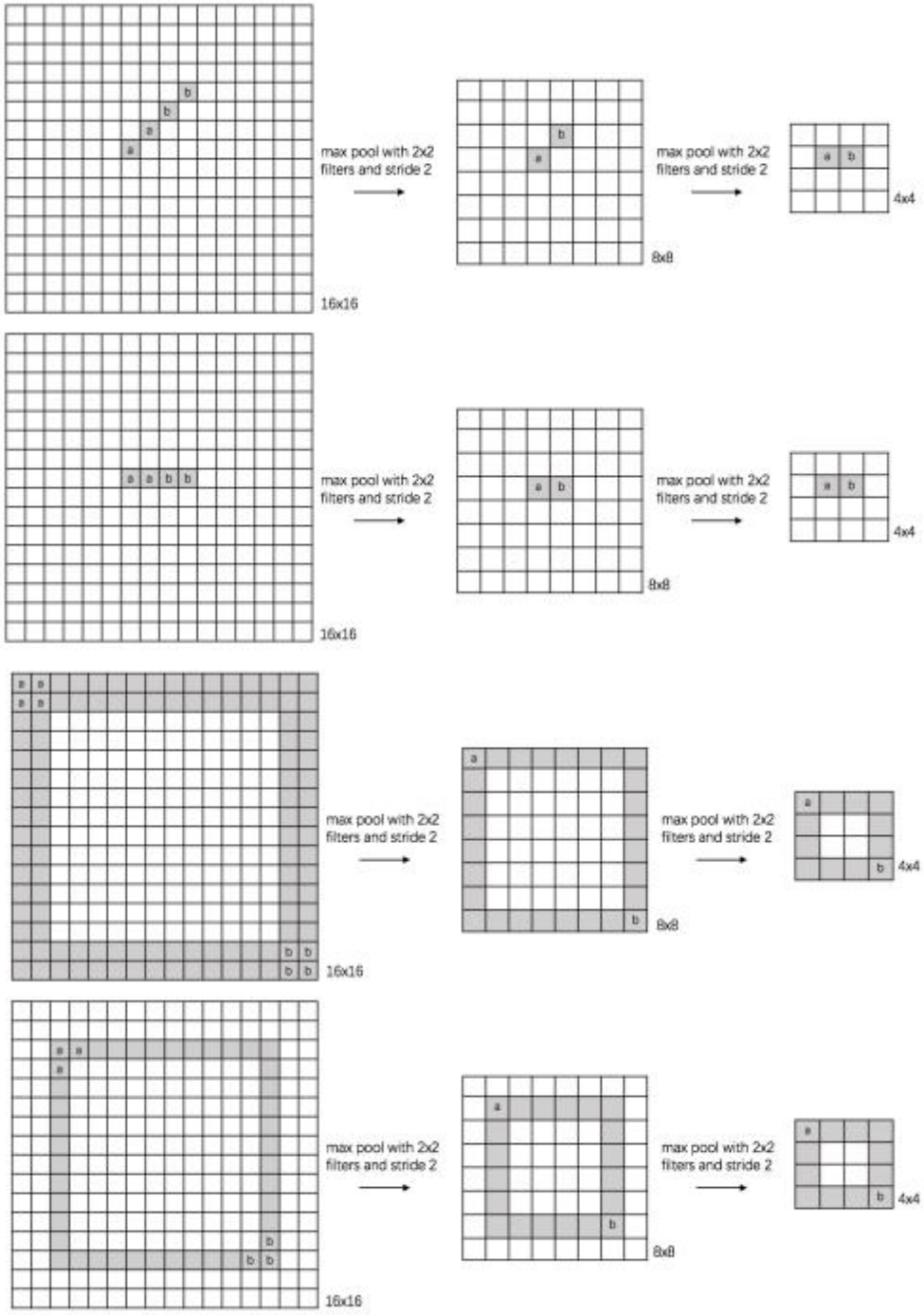
□下采样过程：每邻域n个像素通过池化（pooling）步骤变为一个像素，然后通过标量W_{x+1}加权，再增加偏置b_{x+1}，然后通过一个sigmoid激活函数，产生一个大概缩小为1/n的特征映射图S_{x+1}。

后续研究subsampling/pooling有很多发展

Subsampling/Pooling 的作用

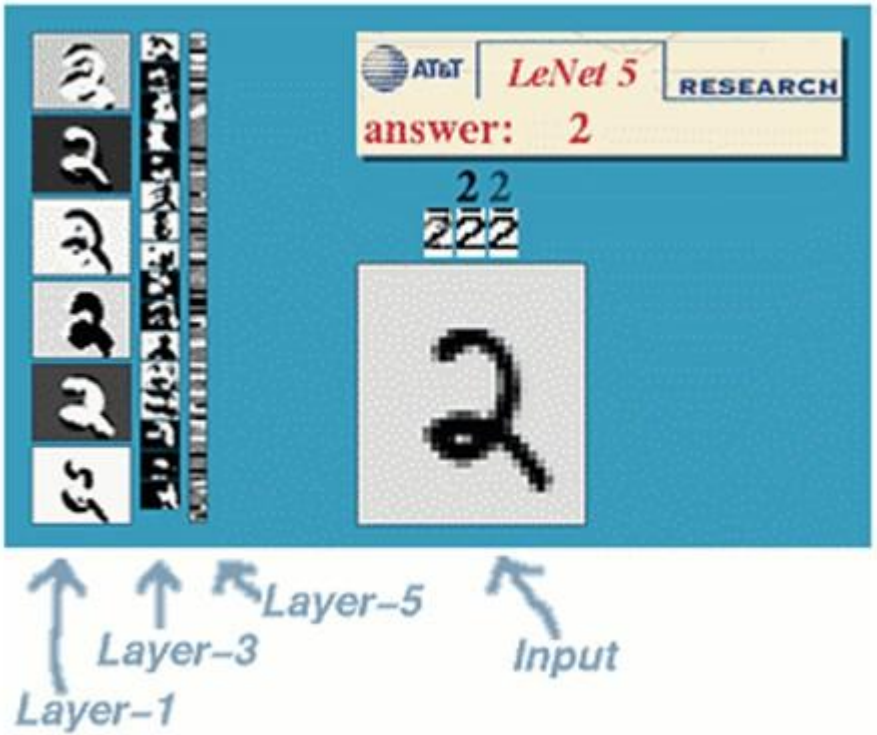
- Subsampling
- Invariance（一定程度上）
- ..... (由于越来越复杂）

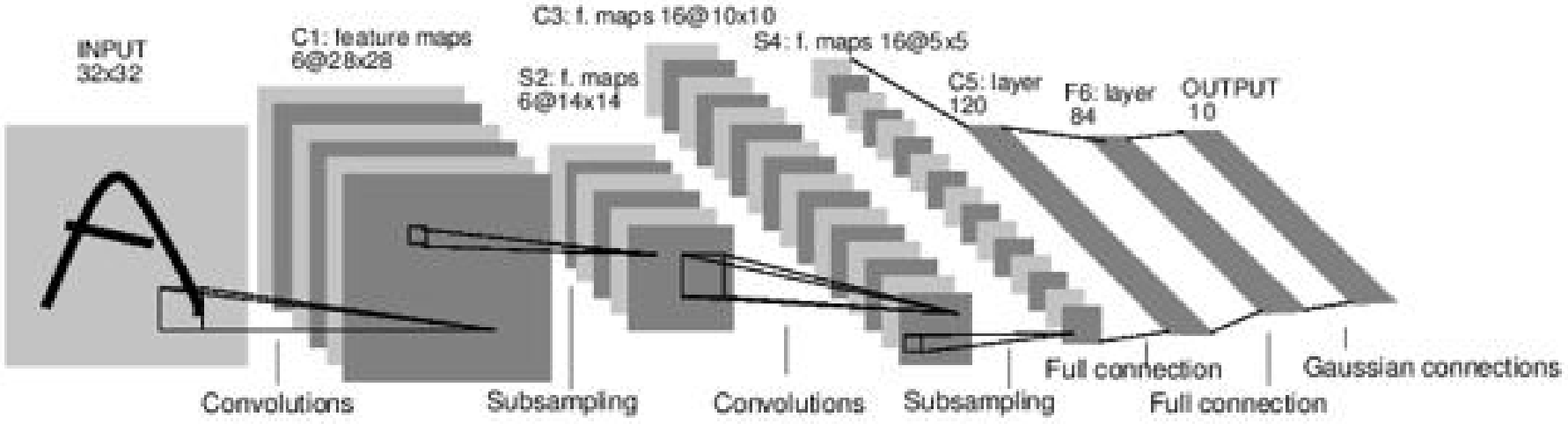




CNN的完整模型

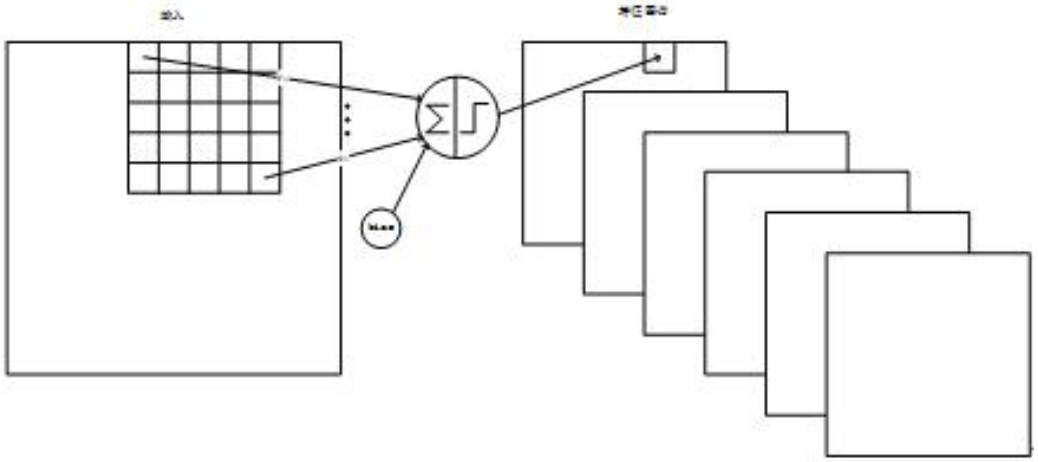
- CNN典型代表：LeNet-5
- 上世纪90年代成功用于支票手写数字识别(>10%US checks)等领域





LeNet-5具体结构

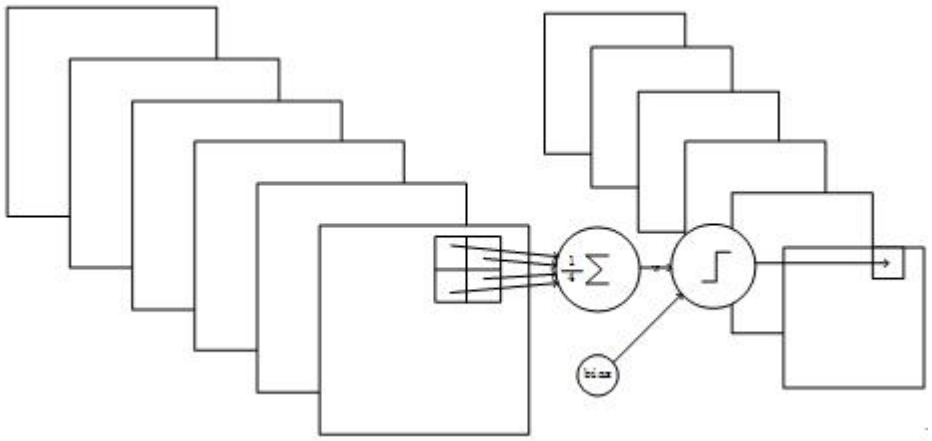
- C1层的结构



C1层是卷积层，形成6个特征图谱。特征图谱中的每个单元与输入层的一个5x5的相邻区域相连，即卷积的输入区域大小是5x5，每个特征图谱内参数共享，即每个特征图谱内只使用一个共同卷积核，卷积核有5x5个连接参数加上1个偏置共26个参数，C1层共有26x6=156个训练参数。卷积区域每次滑动一个像素，这样卷积层形成的特征图谱每个的大小是28x28。

LeNet-5具体结构

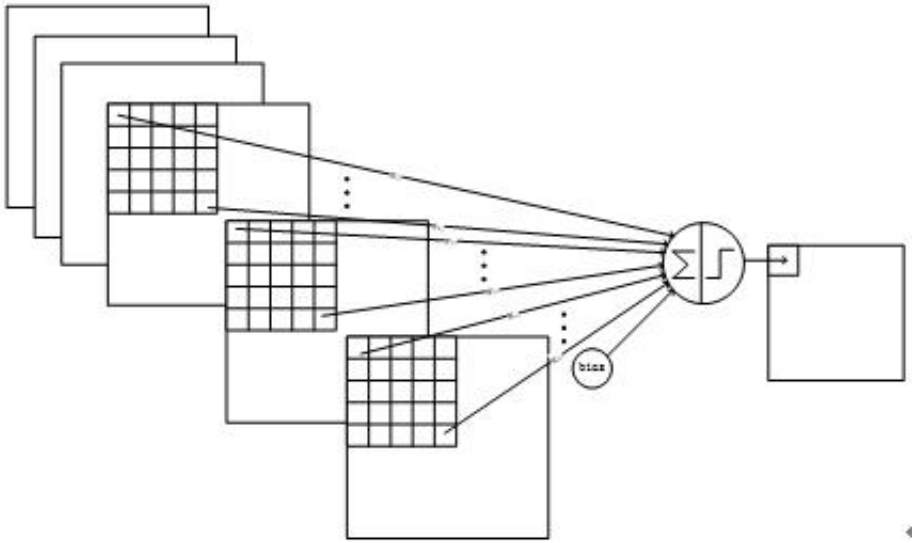
- S2层的网络结构



S2层是一个下抽样层，与C1层一一对应。C1层的6个28x28的特征图谱分别进行以2x2为单位的下抽样得到6个14x14的图。每个特征图谱使用一个下抽样核，每个下抽样核有两个训练参数，所以共有2x6=12个训练参数。

LeNet-5具体结构

- C3层的网络结构

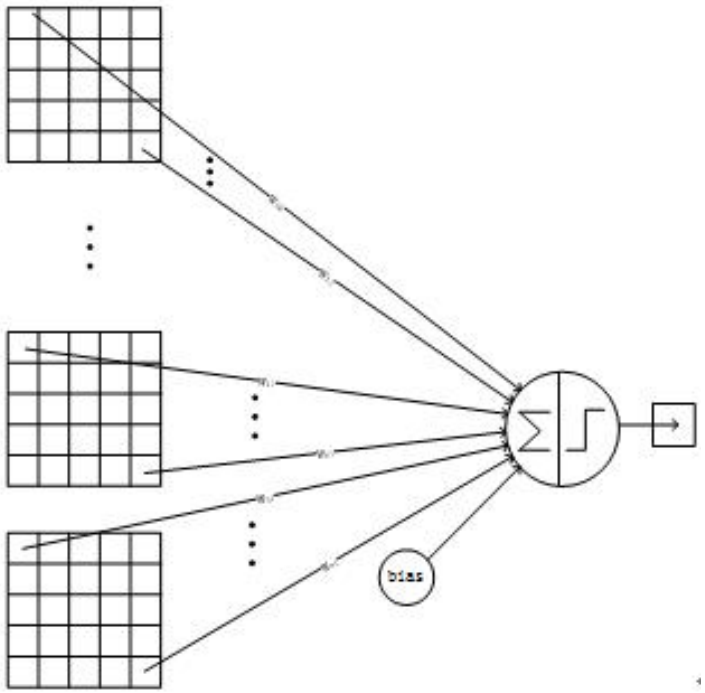


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

C3层是卷积层，卷积核和C1相同，不同的是C3的每个节点与S2中的多个图相连。C3层有16个10x10的图，每个图与S2层的连接的方式如表所示。C3与S2中前3个图相连的卷积结构见图。这种不对称的组合连接的方式有利于提取多种组合特征。有(5x5x3+1)x6 + (5x5x4+1) x6 + (5x5x4+1)x3 + (5x5x6+1)x1 = 1516个训练参数

LeNet-5具体结构

- S4层是一个下采样层。C3层的16个10x10的图分别进行以2x2为单位的下抽样得到16个5x5的图
- C5层是一个卷积层。由于S4层的16个图的大小为5x5，与卷积核的大小相同，所以卷积后形成的图的大小为1x1。这里形成120个卷积结果。每个都与上一层的16个图相连。所以共有(5x5x16+1)x120 = 48120个参数

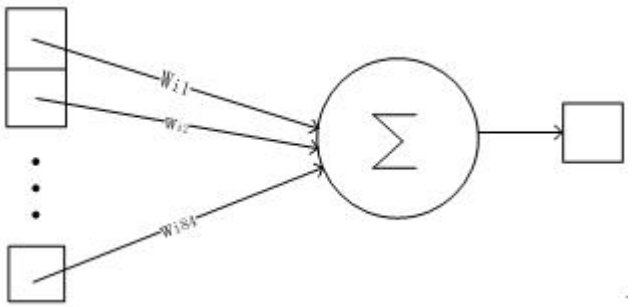


LeNet-5具体结构

- F6 层有84个单元（选这个数字的原因来自于输出层的设计），与C5层全相连
- Output层也是全连接层，共有10个节点，分别代表数字0到9，且如果节点i的值为0，则网络识别的结果是数字i。输出层采用的是径向基函数（RBF）单元。假设x是上一层的输出，y是RBF的输出，则RBF输出的计算方式是：

$$y_i = \sum_j (x_j - w_{ij})^2$$

- $\{w_{ij}\}$ were designed to represent a stylized image of the corresponding character class drawn on a 7x12 bitmap (hence the number 84). The parameter vectors of these units were chosen by hand and kept fixed (at least initially).



CNN训练

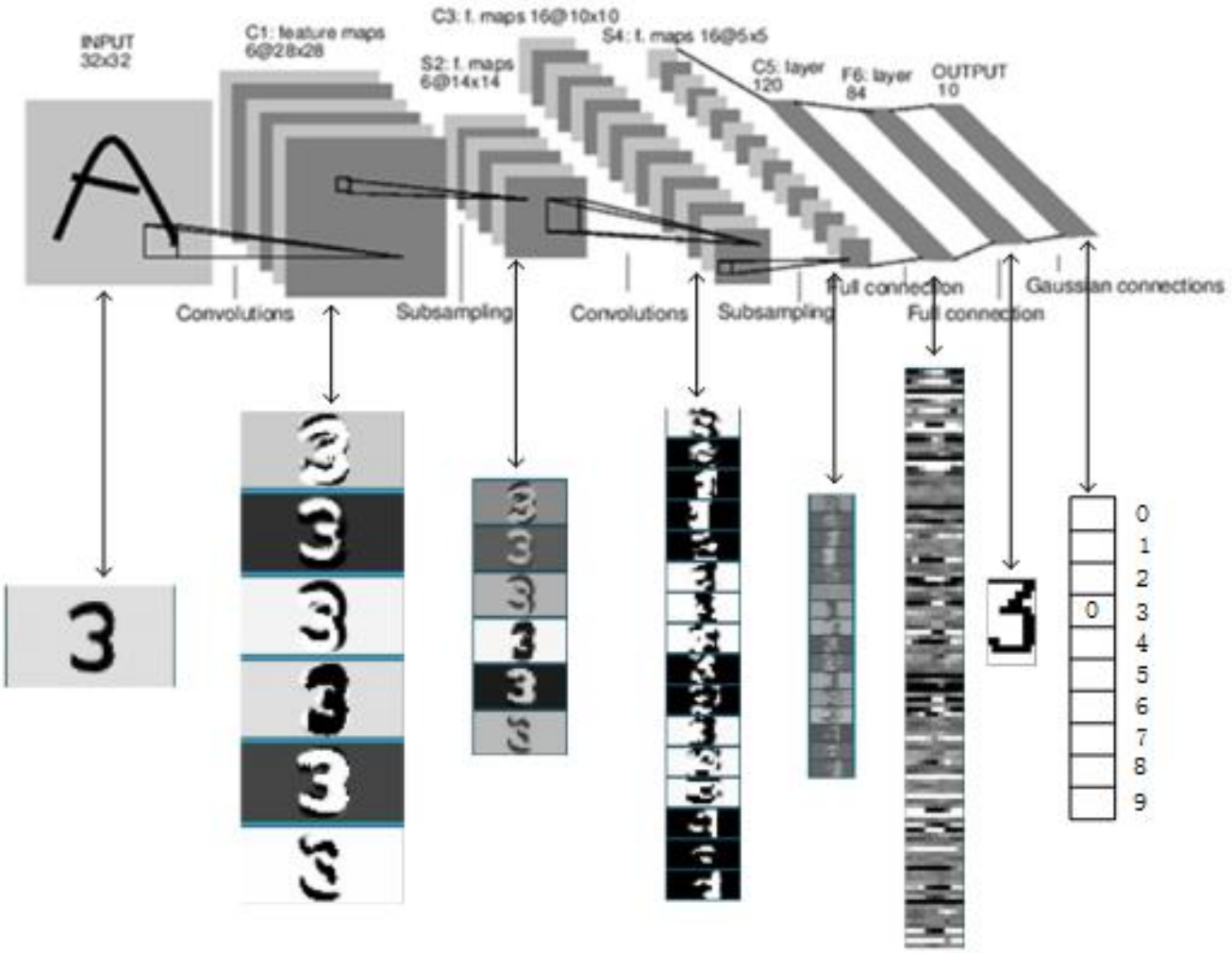
- 用BP算法
- 有一些技巧

  如采用ReLU (Rectified Linear Unit)激活函数来应对梯度消失

  .....
- 建议公式推导一遍，自己编程实现一遍
- Yann LeCun: “在90年代，我被认为是唯一能让卷积网络运转的人（然而事实并非如此）”。

Q10: 1-D CNN & an application，从底层编程实现

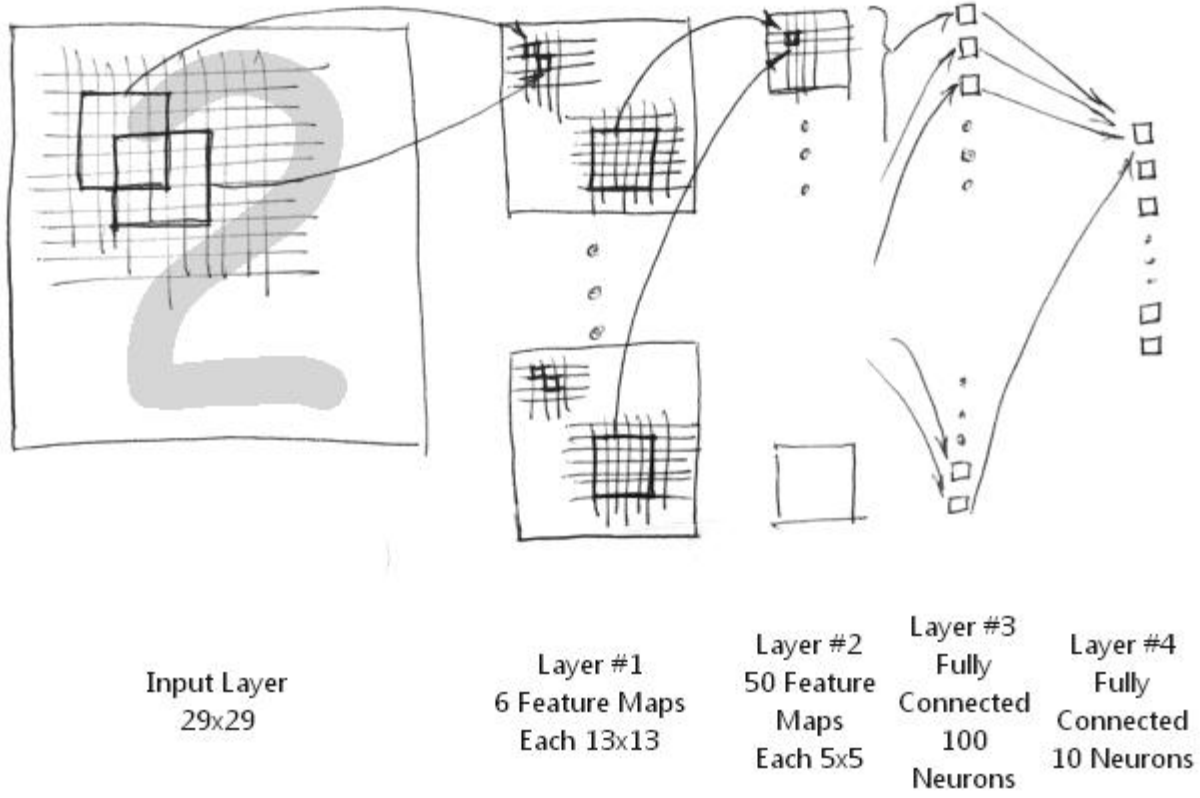
LeNet-5整体计算可视化



- 3D Visualization of CNN
  http://scs.ryerson.ca/~aharley/vis/conv/

简化的LeNet-5

- 简化的LeNet-5系统把下采样层和卷积层结合起来，避免了过多参数的学习，同时保留了对图像位移、扭曲的鲁棒性。



Generative Adversarial Networks

生成式对抗网络

对抗样本

- Adversarial Examples的概念由C. Szegedy 等人在ICLR2014发表的Intriguing properties of neural networks 中提出，即在数据集中通过故意添加细微的干扰形成输入样本，受干扰之后的输入导致模型以高置信度给出了一个错误的输出。他们发现包括CNN在内的深度学习模型对于对抗样本都具有极高的脆弱性。从而将矛头直指深度学习。
- Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, by Nguyen A, et al, CVPR 15一文更是发现，面对一些人类完全无法识别的样本，深度学习模型居然会以高置信度将它们进行分类。

原始图像以 60%的置信度将它归为“熊猫”。

+

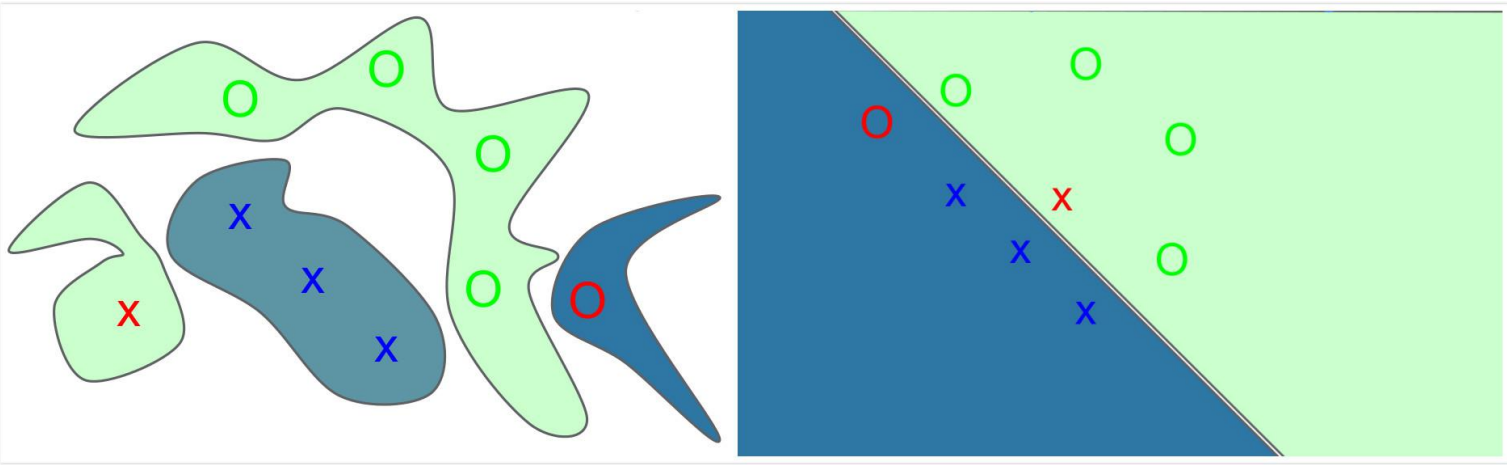
微小的对抗性干扰

=

不易察觉地修改的图像，以 99%的置信度将它归为长臂猿。

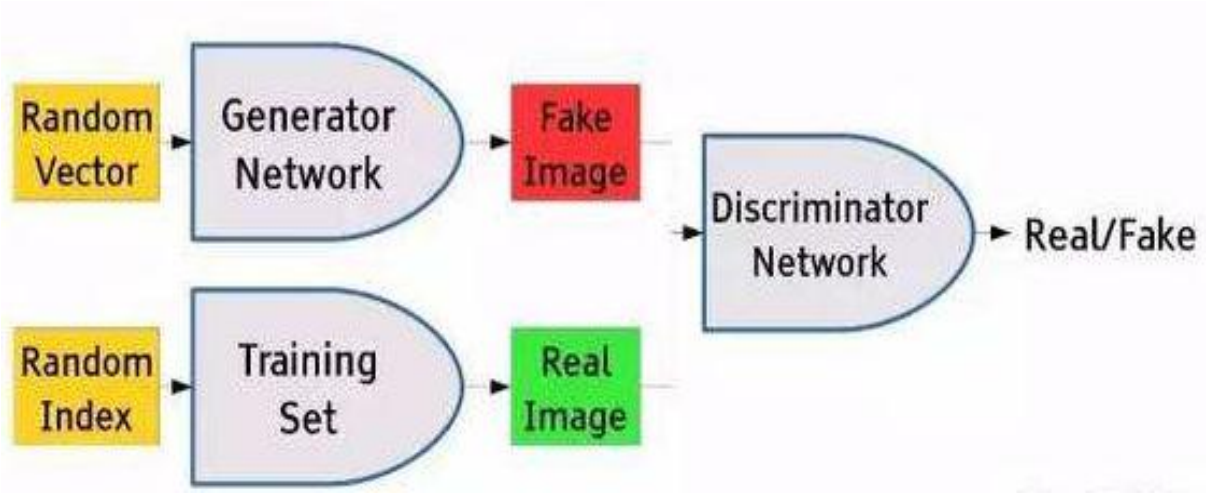
对抗样本

- 论文 Exploring the Space of Adversarial Images指出，至少在图像空间中，对抗图片绝非孤立点，而是占据了很大一部分空间
- 一些人开始认为深度学习不是deep learning, 而是deep flaw(错误)
- Lipton的 (Deep Learning's Deep Flaws)'s Deep Flaws 指出，事实上对于对抗样本的脆弱性并不是深度学习所独有的，在很多的机器学习模型中普遍存在，因此进一步研究有利于抵抗对抗样本的算法实际上有利于整个机器学习领域的进步。
- 造成对抗样本的原因是什么呢？一个推断性的解释是深度神经网络的高度非线性特征，以及过拟合导致的。深度学习可解释性差。
- Ian Goodfellow 在ICLR2015年的论文中，发现线性模型也对对抗样本表现出明显的脆弱性。不仅过拟合，欠拟合也会。

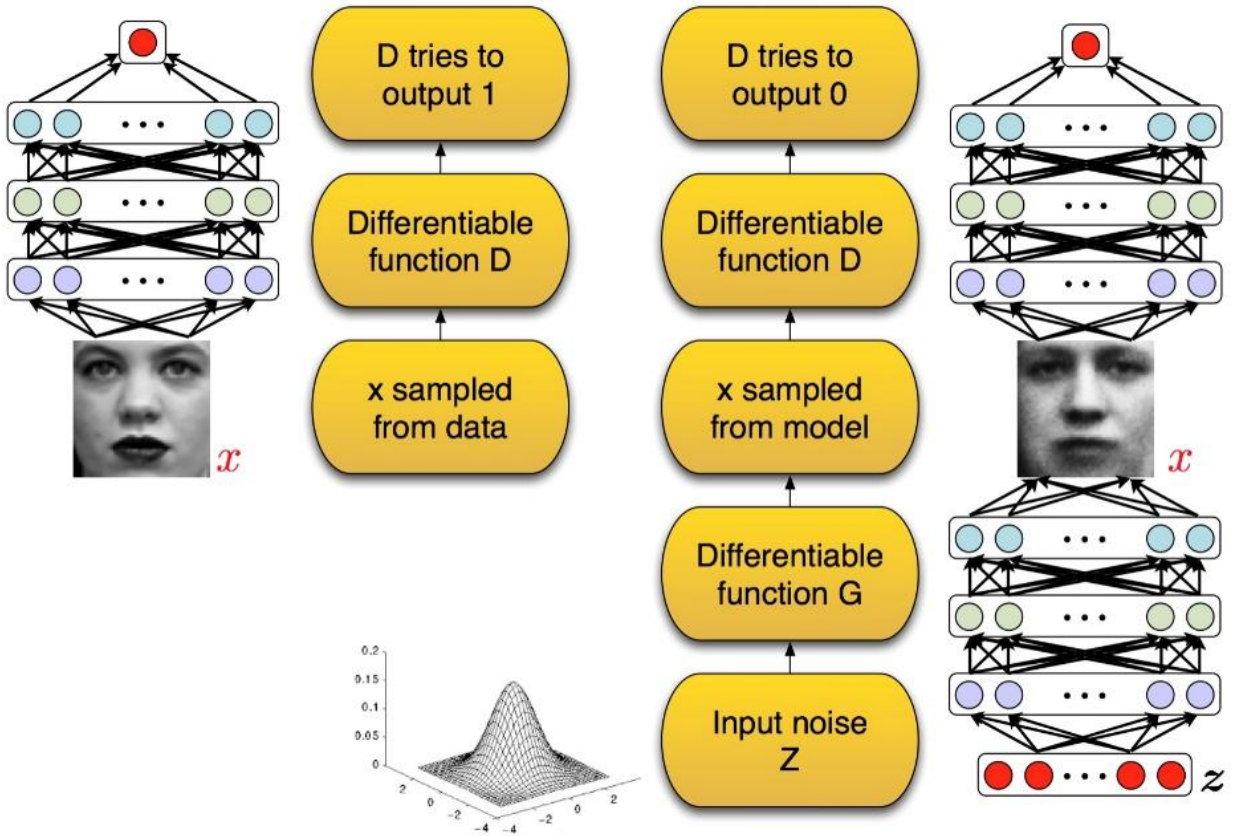


Generative Adversarial Networks

- 随着对对抗样本的更多更深入研究，人们逐渐发现，对抗样本并不只是诅咒，也可能是祝福，因为可以利用对抗样本。Ian Goodfellow等人在NIPS2014提出了Generative Adversarial Nets。
- Yann LeCun: 短期看，深度学习领域有许多有意思的进展，也许数量太多，有少数几个ideas引起了我的注意。生成对抗式网络，以及现在被提出的一些变体，是深度学习领域过去10年我认为最有意思的idea。



- GAN 包括生成网络（generator）和判别网络（discriminator）。生成网络 G 捕捉样本数据的分布，用服从某一分布（均匀分布，高斯分布等）的噪声 z 生成一个类似真实训练数据的样本，追求效果是越像真实样本越好；判别网络 D 是一个二分类器，估计一个样本来自于训练数据（而非生成数据）的概率，如果样本来自于真实的训练数据D输出1，否则D 输出0。
- 生成网络 G 好比假币制造团伙，专门制造假币，判别网络 D 好比警察，专门检测使用的货币是真币还是假币，G 的目标是想方设法生成和真币一样的货币，使得 D 判别不出来；D 的目标是想方设法检测出来 G 生成的假币。



Generative Adversarial Networks

- 在训练的过程中固定一方，更新另一方的网络权重，交替迭代，在这个过程中，双方都极力优化自己的网络，从而形成竞争对抗，直到双方达到一个动态的平衡，此时生成模型 G建模了训练数据的分布（造出了和真实数据一模一样的样本），判别网络再也判别不出来结果，准确率为 50%，约等于乱猜。
- 目标函数：

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

当固定生成网络 G 的时候，对于判别网络 D 的优化，可以这样理解：输入来自于真实数据，D 优化网络结构使自己输出 1，输入来自于生成数据，D 优化网络结构使自己输出 0；当固定判别网络 D 的时候，G 优化自己的网络使自己输出尽可能和真实数据一样的样本，并且使得生成的样本经过 D 的判别之后，D 输出高概率。

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.

GAN 尽管用 z 作为输入，但生成模型如何利用这个 z呢？GAN 的学习模式太过于自由了，使得 GAN 的训练过程和训练结果很多时候都不太可控。为了稳定 GAN，后来的研究者们分别提出了许多训练技巧和改进方法。

比如在原始 GAN 论文中，每次学习参数的更新过程，被设为 D 更新 k 回， G 才更新 1 回，就是有些出于减少 G 的“自由度”的考虑。

Generative Adversarial Networks

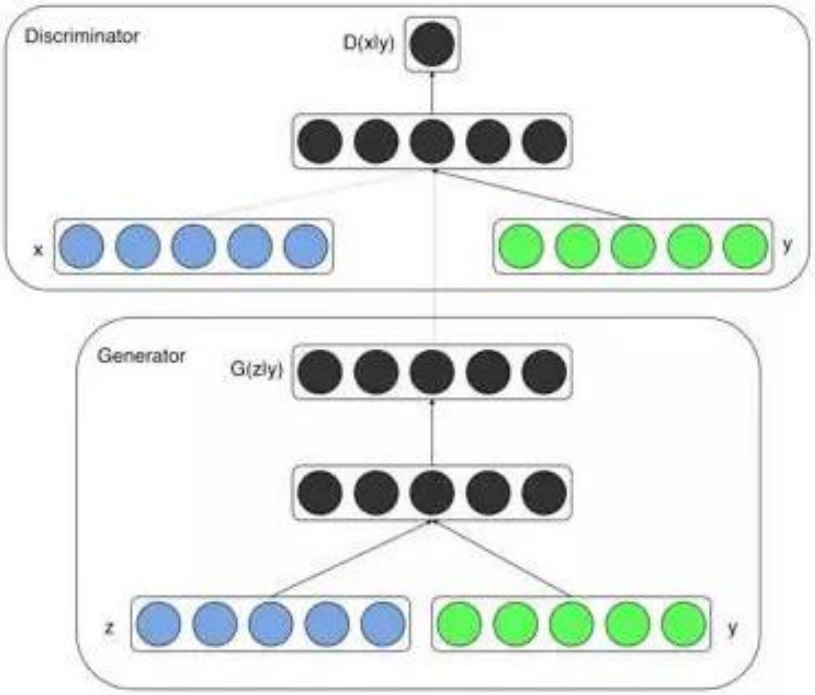
- “**判别网络**必须要发展出一个好的数据内部表征，来有针对性地解决问题。训练后，判别网络还可以被当成分类器中的一个特征提取器”---LeCun访谈
- **生成网络**：生成（如下几页）

MNIST 手写数据集上生成的结果，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

Conditional Generative Adversarial Networks

- 想法很简单，就是给 GAN 加上条件，让生成的样本符合我们的预期，这个条件可以是类别标签（例如 MNIST 手写数据集的类别标签），也可以是其他的多模态信息（例如对图像的描述语言）等

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]. \quad (2)$$



Conditional Generative Adversarial Networks

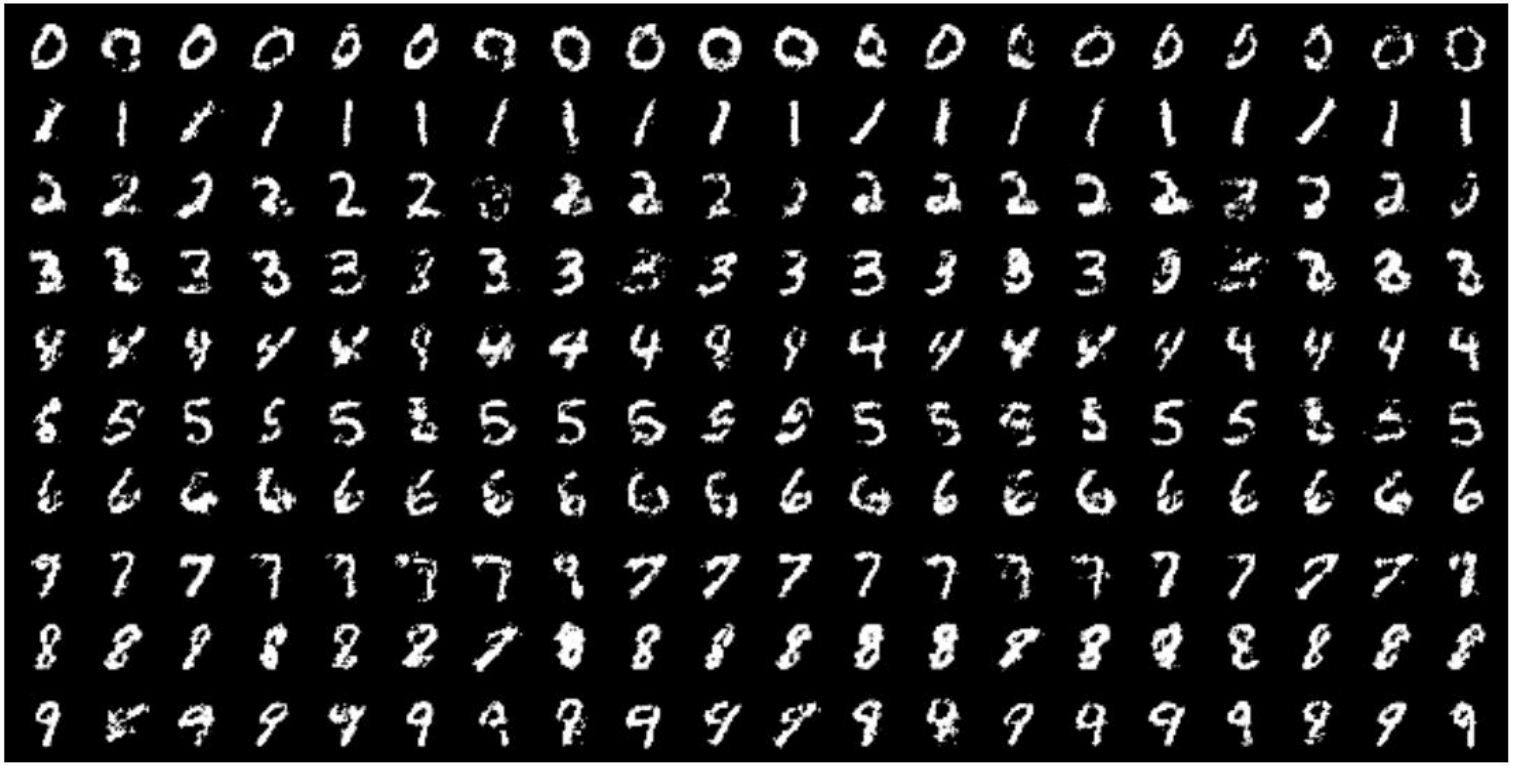
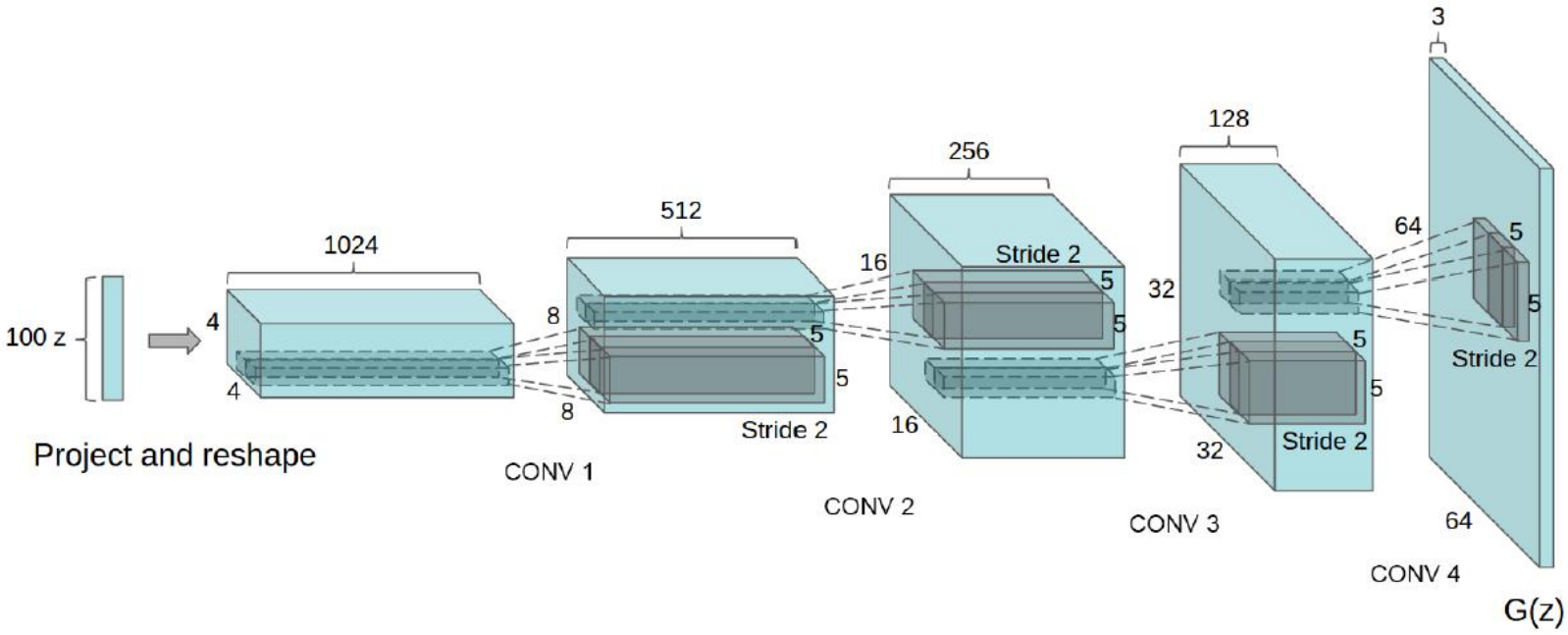


Figure 2: Generated MNIST digits, each row conditioned on one label

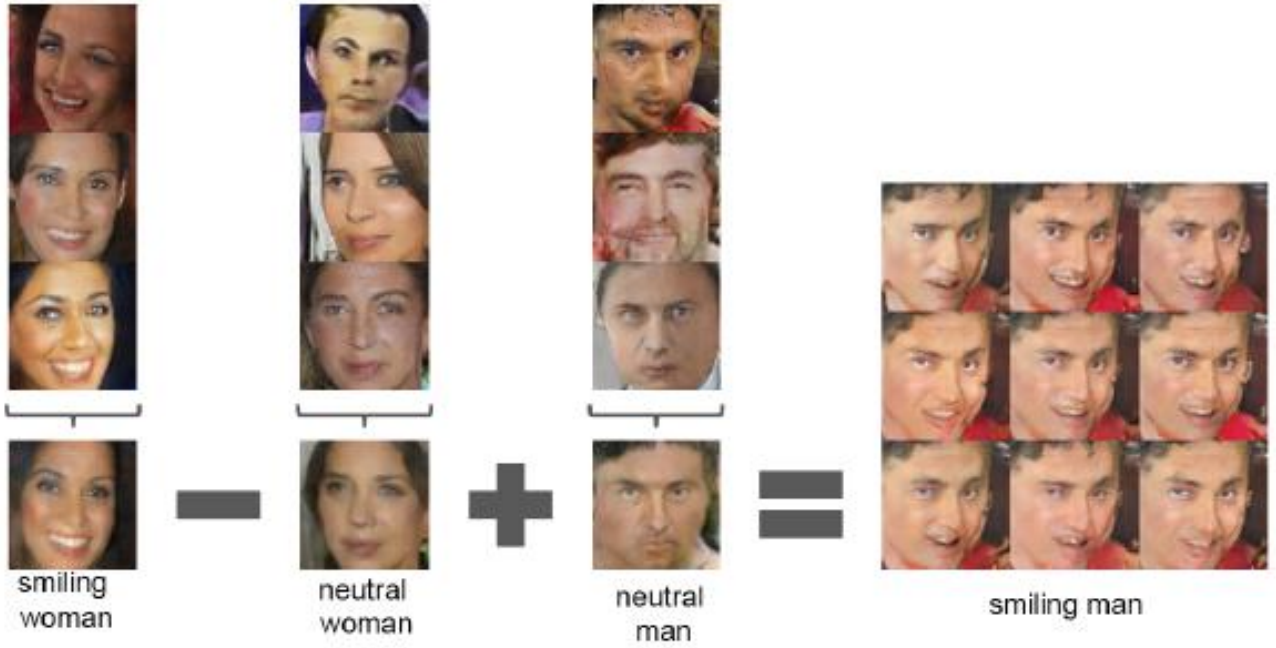
Deep Convolutional Generative Adversarial Networks



Alec Radford, Luke Metz, Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv: 1511.06434, 2015

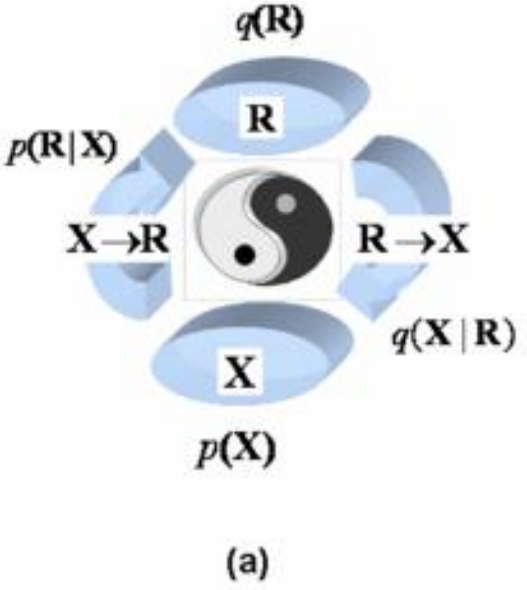
Deep Convolutional Generative Adversarial Networks

Generated bedrooms


smiling woman
neutral woman
neutral man
smiling man
man with glasses
man without glasses
woman without glasses
woman with glasses

两个学习器

- Predictability Minimization (Juergen Schmidhuber, 1992)
- Ying-Yang Machine (BYY harmony learning), CUHK Lei XU



阴阳：系统构成互补观

(b) 老子《道德经》
“万物负阴而抱阳，冲气以为和”

(c) 宋邵雍《观物外篇》
“阳以阴为基”“阴以阳为唱”

(d) 周易《系辞上》“乾以易知，坤以简能”
《周易集解》虞翻曰
“陽見稱易，陰藏為簡，簡，閱也”

和谐：系统寻优根本道

(e) 《正蒙》“太和所谓道”

(f) 阴阳以最默契或交换信息最小的方式达到最大一致。

- .....

Deep SVM

- DNN提取特征作为SVM输入
- DNN+SVM联合训练
- 以SVM的激活函数(max-margin)为目标函数训练DNN

Q11: Deep SVM & an application，不希望是第一种思想的

Deep Forest

Q12: Deep Forest & an application

Q13: 自选

Topics

- **Unsupervised learning**
  - 无监督学习很重要, 过去发展比较缓慢，做clustering, PCA, VQ之类
  - Autoencoder 把无监督转化为监督
- **Recurrent neural network (RNN)**

  尽管前馈网络有难以置信的成功，它们受制于无法明确模拟时间关系，以及所有数据点都是由固定长度的向量组成的假设。

  Long-Short Term Memory (LSTM)

  Bidirectional RNN
- Transfer learning, Reinforcement learning
- **领域应用**

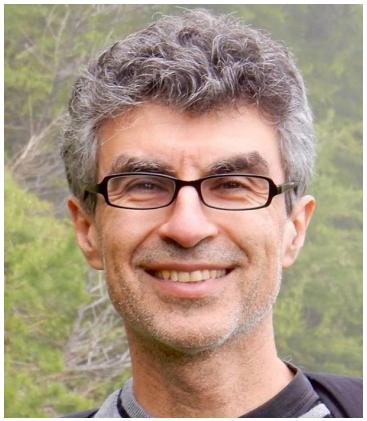
  视频、医疗、金融 .....

Some important contributors

Geoffrey Hinton --> Yann LeCun <--> Yoshua Bengio --> Ian Goodfellow

postdoc

AT&T Colleague



M.I. Jordan's postdoc

student

Juergen Schmidhuber

NIPS 2016, Ian Goodfellow的GAN Tutorial

如何不查文献知道一个NN方面的想法是不是新的？

一些评论

- IEEE Spectrum：这些天我们看到了许多关于深度学习的新闻。在这些对深度学习的众多描述中，你最不喜欢哪一种？
- Yann LeCun：我最不喜欢的描述是“它像大脑一样工作”。我不喜欢人们这样说的原因是，虽然深度学习从生命的生物机理中获得灵感，但它与大脑的实际工作原理差别非常非常巨大。将它与大脑进行类比给它赋予了一些神奇的光环，这种描述是危险的，这将导致天花乱坠的宣传，大家在要求一些不切实际的事情。人工智能之前经历了几次寒冬就是因为人们要求了一些人工智能无法给予的东西。

Mathematics is the art of giving the same name to many different things.

—— Henry Poincare