人工智能原理与技术

二、神经网络基础

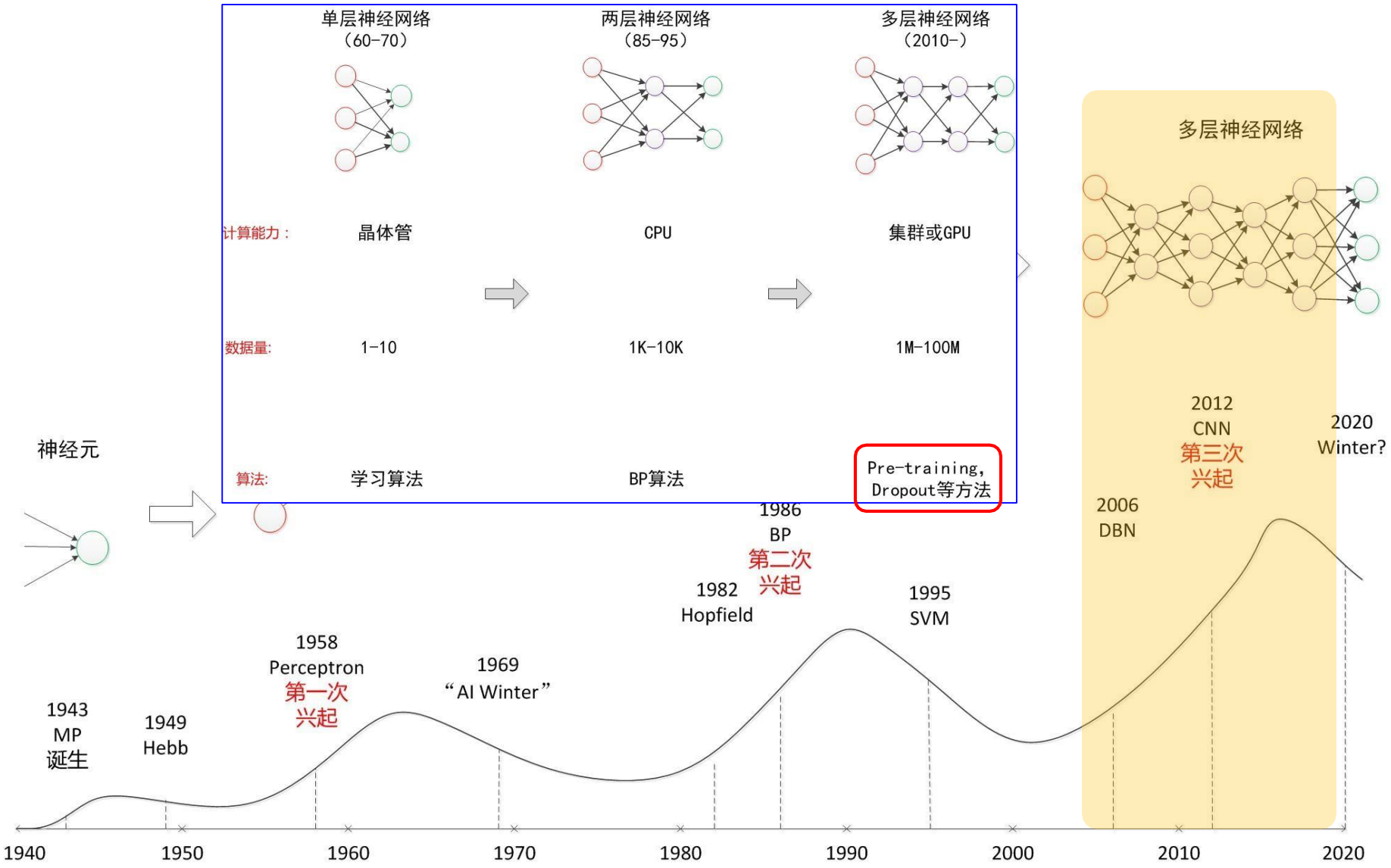
2. 从神经网络到深度学习

科研大师也看风水，要改名字

Hinton & Deep Learning

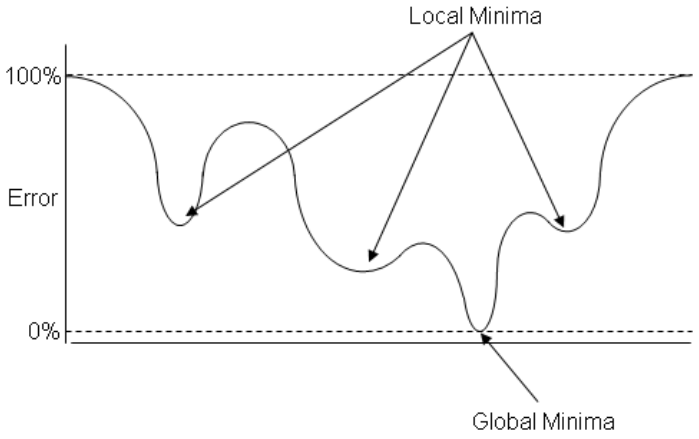
- 2003年, Geoffrey Hinton, 还在多伦多大学, 在神经网络的领域苦苦坚守
- 2003年在温哥华大都会酒店, 以Hinton 为首的十五名来自各地的不同专业的科学家, 和加拿大先进研究院 (Canadian Institute of Advanced Research, CIFAR) 的基金管理负责人 Melvin Silverman 交谈
  - Silverman 问大家, 为什么 CIFAR 要支持他们的研究项目
  - 计算神经科学研究者, Sebastian Sung (现为普林斯顿大学教授) 回答道： "喔, 因为我们有点古怪. 如果CIFAR 要跳出自己的舒适区, 寻找一个高风险, 极具探索性的团体, 就应当资助我们了!"
  - 最终 CIFAR 同意从2004年开始资助这个团体十年, 总额一千万加元. CIFAR 成为当时世界上唯一支持神经网络研究的机构
- Hinton 拿到资金支持不久, 做的第一件事, 就是把"神经网络" 改名换姓为 "深度学习"
- 此后，Hinton 的同事不时会听到他突然在办公室大叫： "我知道人脑是如何工作的了!".

神经网络的“第三次起”


单层神经网络
（60-70）
两层神经网络
（85-95）
多层神经网络
（2010-）
计算能力：
晶体管
CPU
集群或GPU
数据量:
1-10
1K-10K
1M-100M
算法:
学习算法
BP算法
Pre-training,
Dropout等方法
多层神经网络
神经元
1943
MP
诞生
1949
Hebb
1958
Perceptron
第一次
兴起
1969
“AI Winter”
1982
Hopfield
1986
BP
第二次
兴起
1995
SVM
2006
DBN
2012
CNN
第三次
兴起
2020
Winter?
1940
1950
1960
1970
1980
1990
2000
2010
2020

逐层预训练 (layer-wise pre-training)

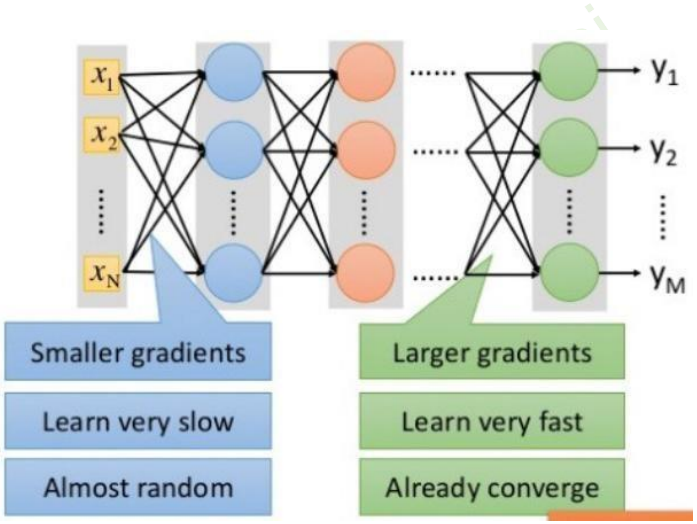
局部极小值



前面说，神经网络如果层数越多，就特别容易出现特别多的局部极小值

问题1：如果层数一直增多，特别容易落到一个比较差的局部极小值里，网络收敛就比较差

梯度消失



问题2：层数多时，容易出现一个梯度消失

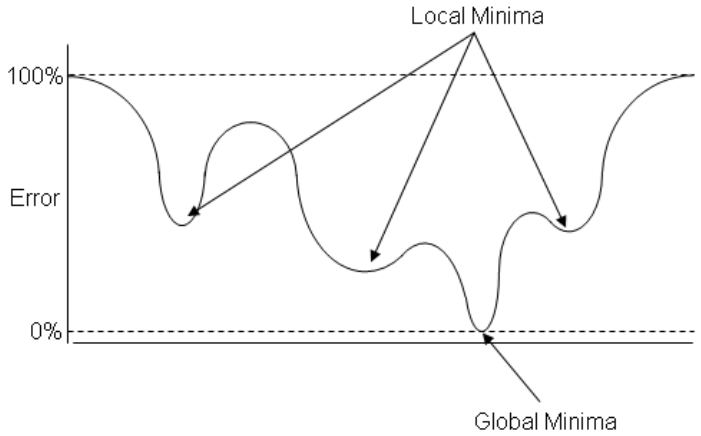
解决方法就是，寻找一个不错的初始值

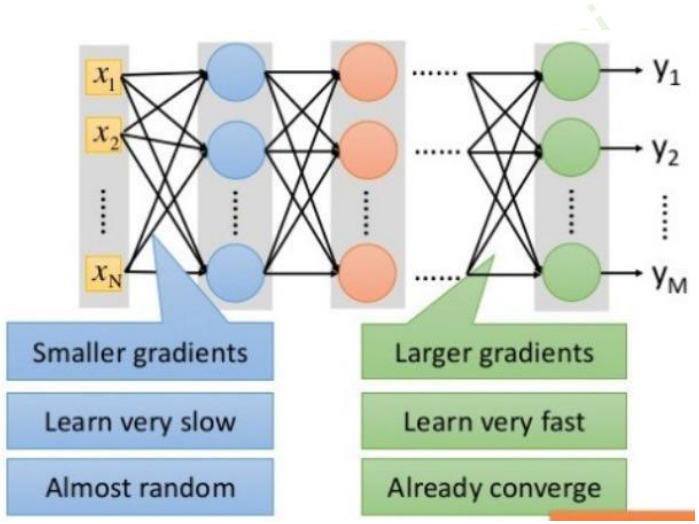
部分图片取自@李宏毅老师课件

逐层预训练 (layer-wise pre-training)

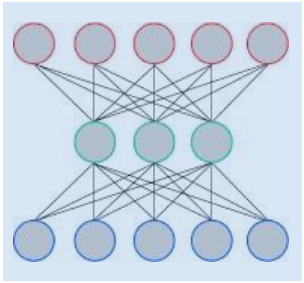
局部极小值

梯度消失





权重初始化

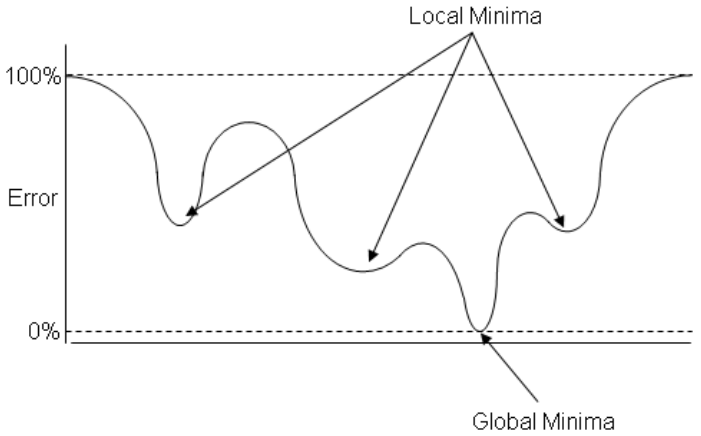


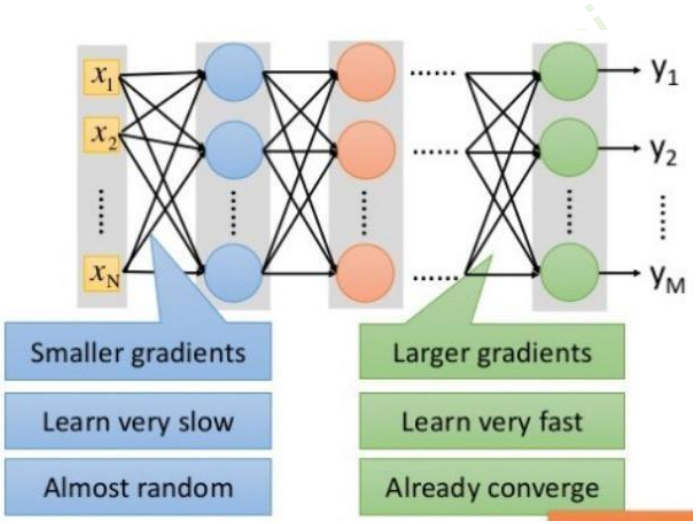
部分图片取自@李宏毅老师课件

逐层预训练 (layer-wise pre-training)

局部极小值

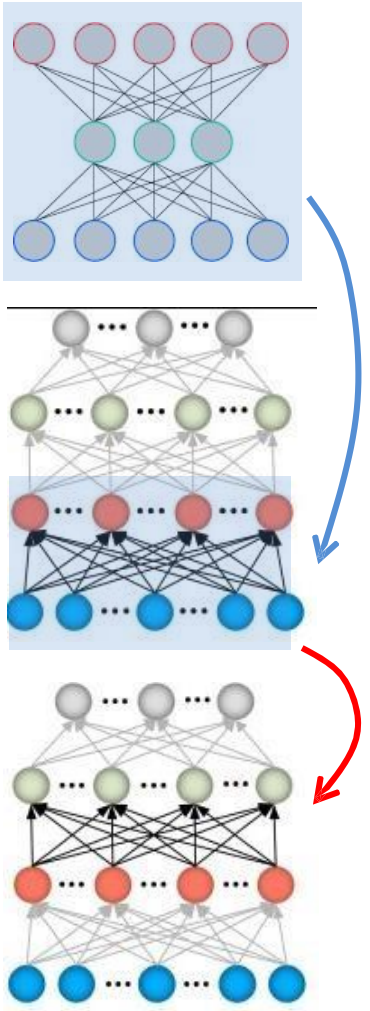
梯度消失





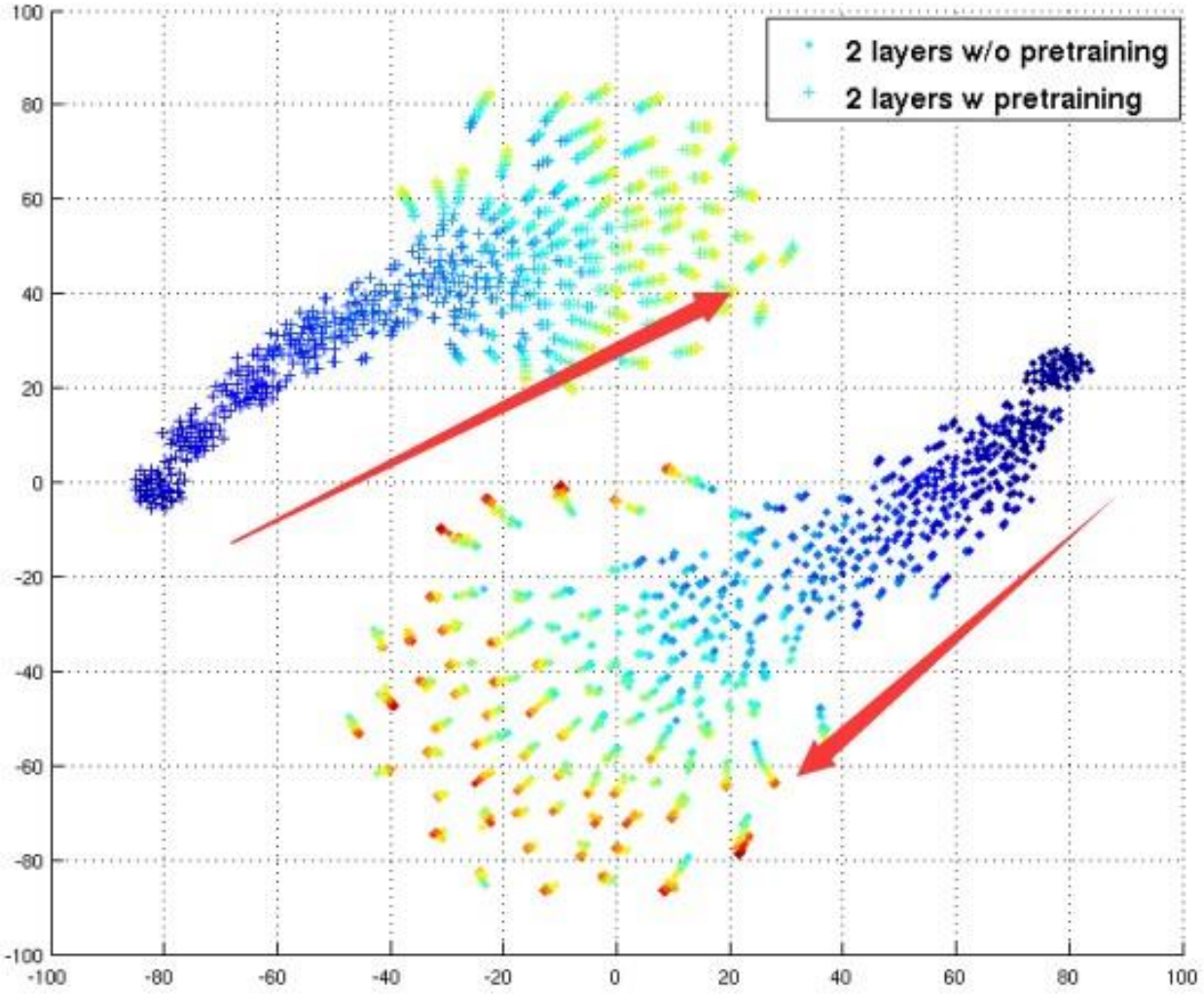
部分图片取自@李宏毅老师课件

权重初始化

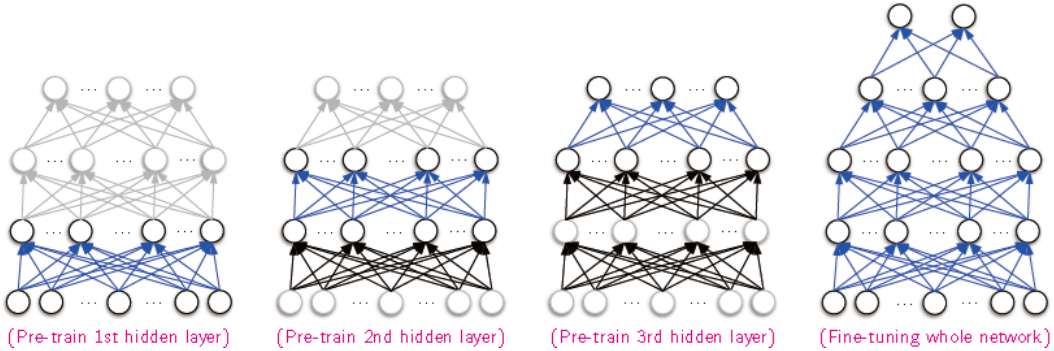


逐层预训练

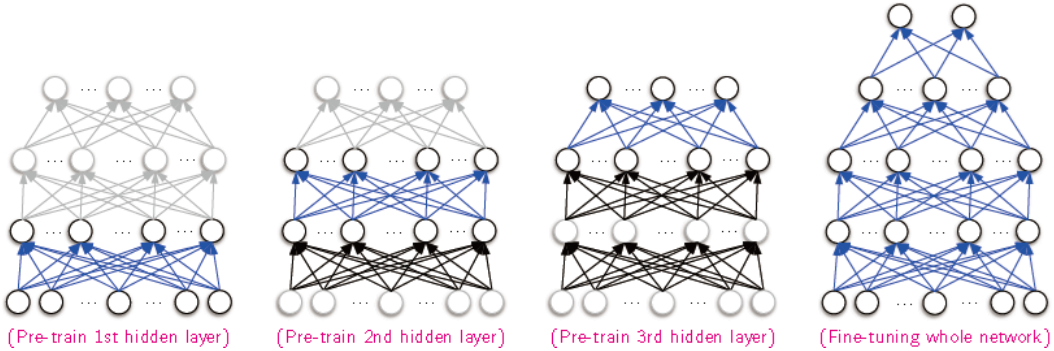
逐层预训练 (layer-wise pre-training)



逐层预训练 (layer-wise pre-training)

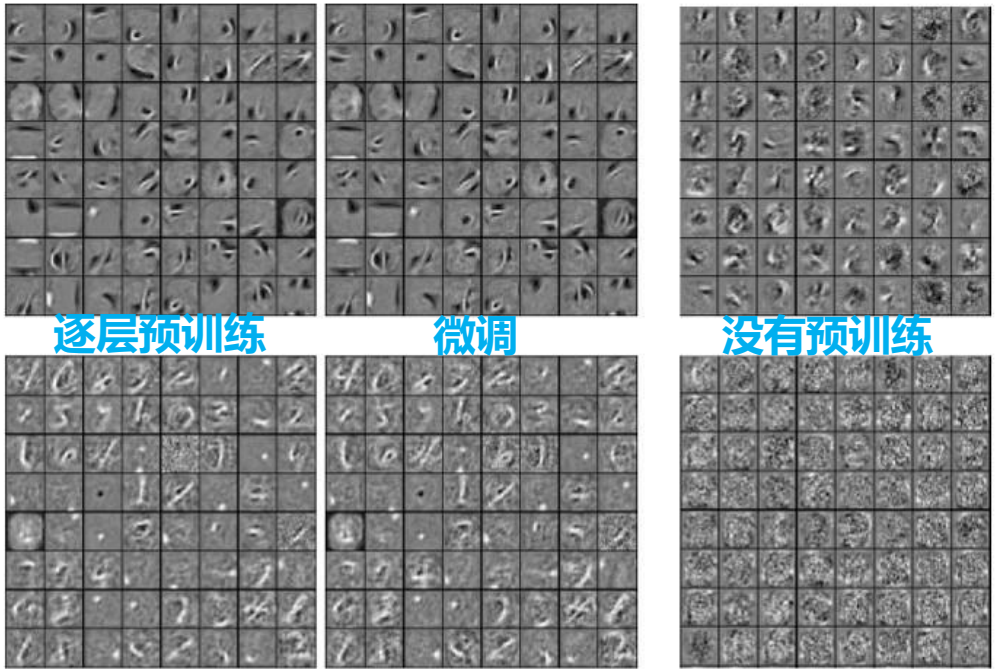


逐层预训练 (layer-wise pre-training)

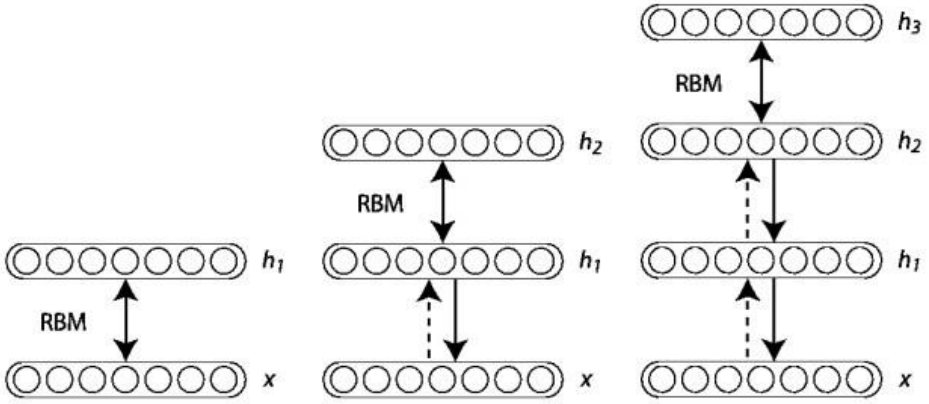


逐层预训练

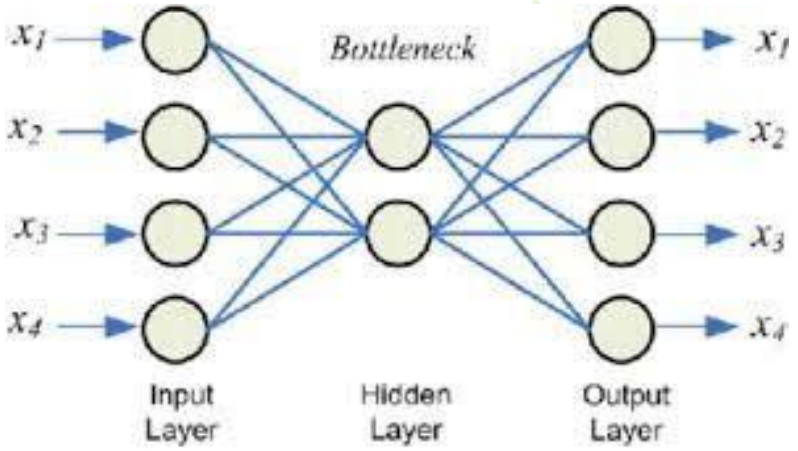
微调(fine-tuning)



受限玻尔兹曼机和自编码器

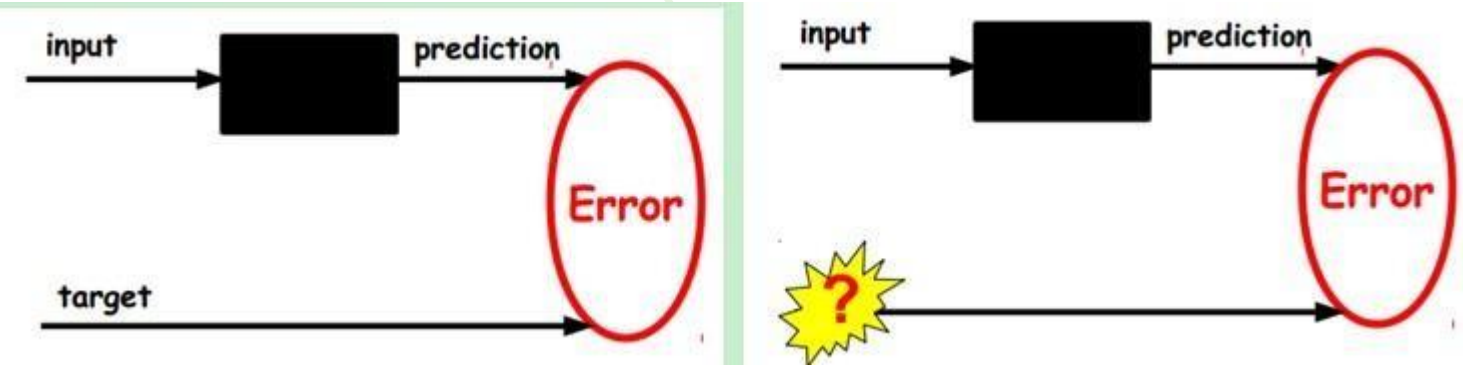


受限玻尔兹曼机 RBM

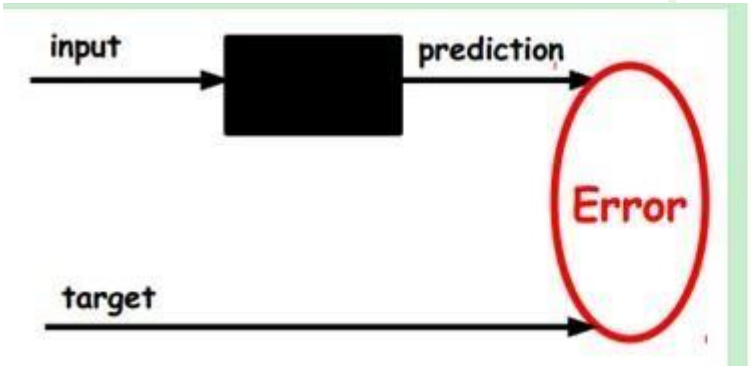


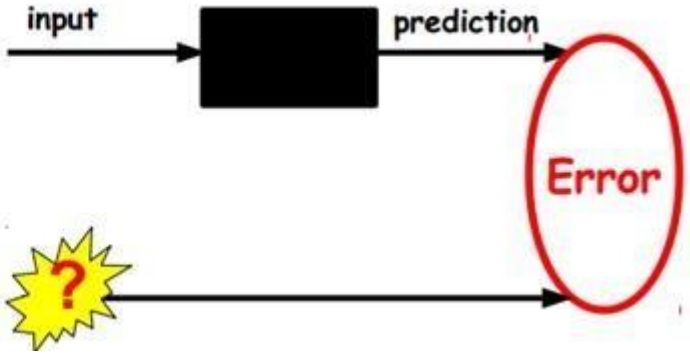
自编码器 Autoencoder

自编码器

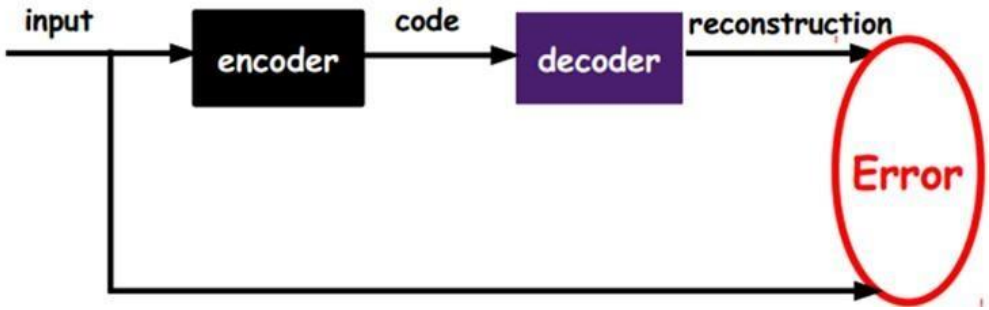


自编码器





- 自编码器(autoencoder)假设输出与输入相同(target=input)，是一种尽可能复现输入信号的神经网络.
- 将input输入一个encoder编码器，就会得到一个code；加一个decoder解码器，输出信息。
- 通过调整encoder和decoder的参数，使得重构误差最小
- 没有额外监督信息：无标签数据，误差的来源是直接重构后信号与原输入相比得到



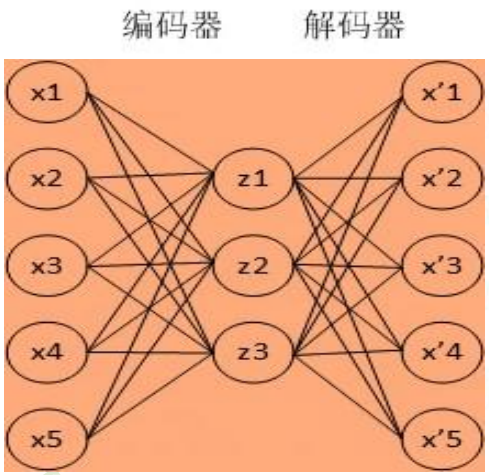
自编码器

- 自编码器一般是一个多层神经网络（最简单：三层）：
  - 训练目标是使输出层与输入层误差最小；
  - 中间隐层是代表输入的特征，可以最大程度上代表原输入信号

▸ 自编码器的学习目标是最小化重构错误:

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - g(f(\mathbf{x}^{(i)}))\|^2$$

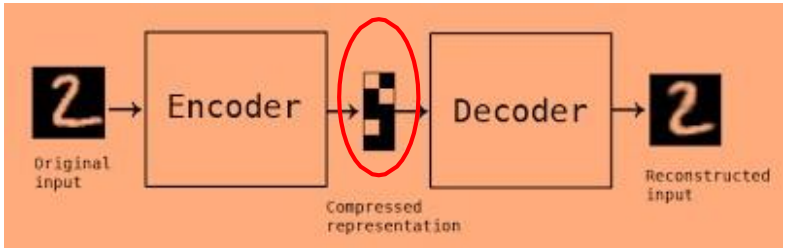
$$= \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - f \circ g(\mathbf{x}^{(i)})\|^2$$



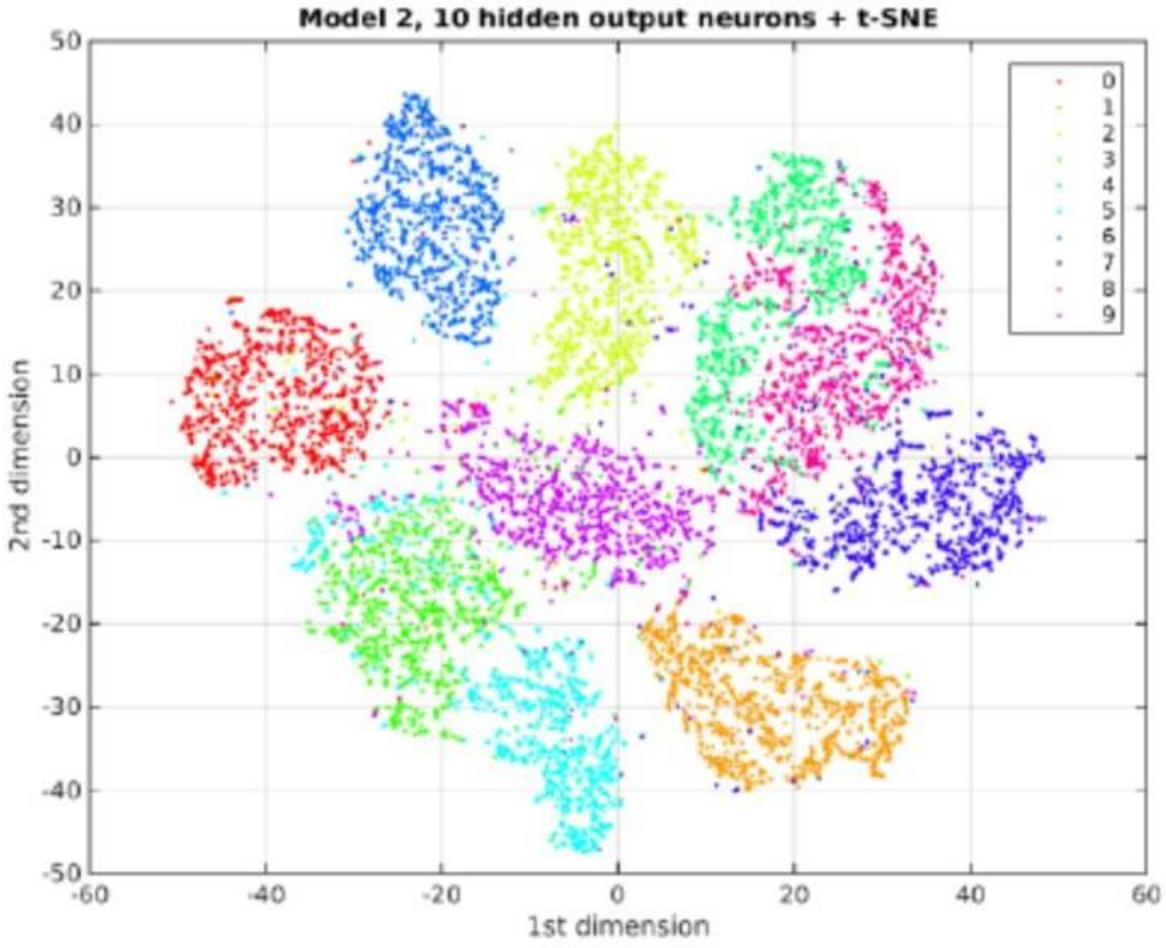
▸ 编码器（Encoder） $f : \mathbb{R}^d \to \mathbb{R}^{d'}$

▸ 解码器（Decoder） $g : \mathbb{R}^{d'} \to \mathbb{R}^d$.

自编码器

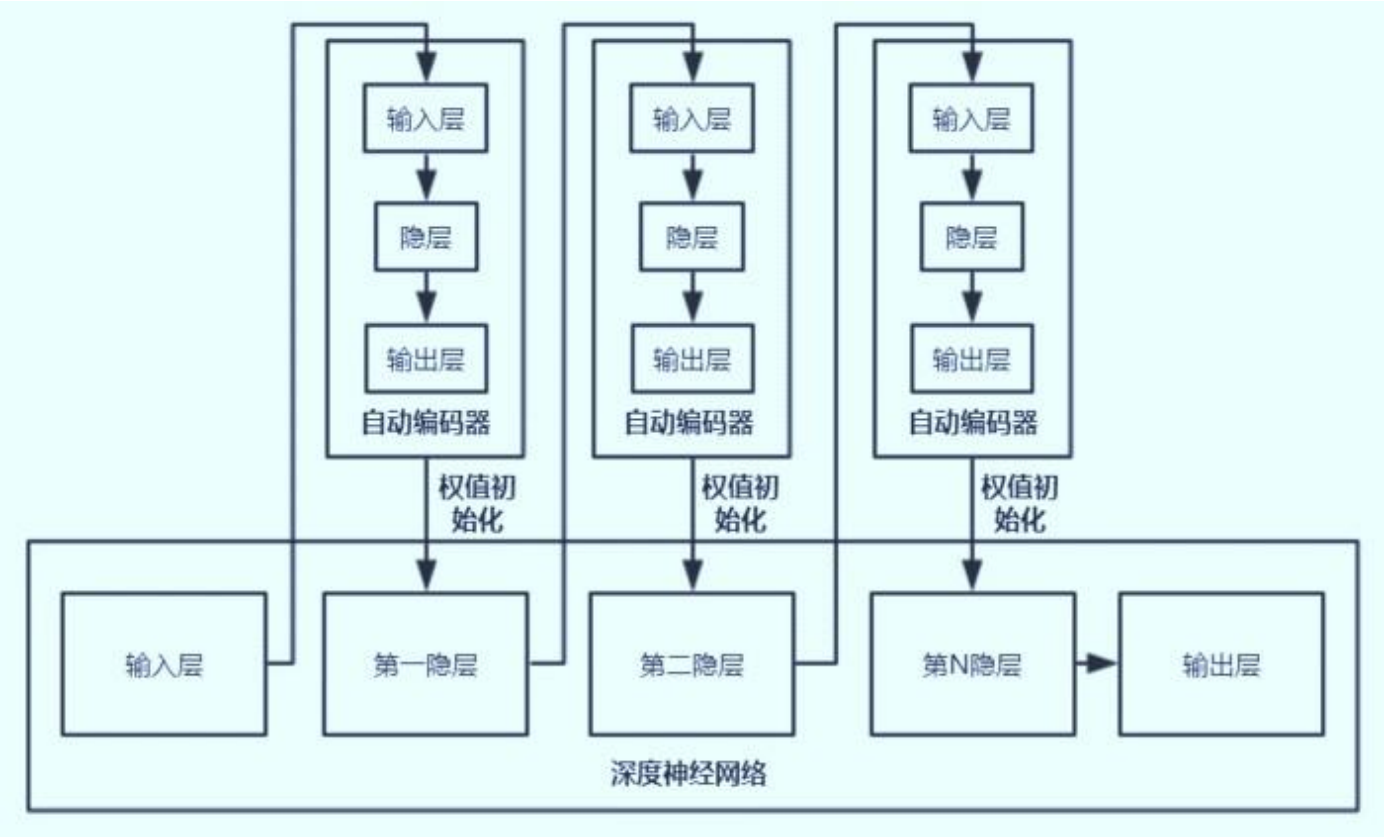


- 很早就被提出了，最初用来降维



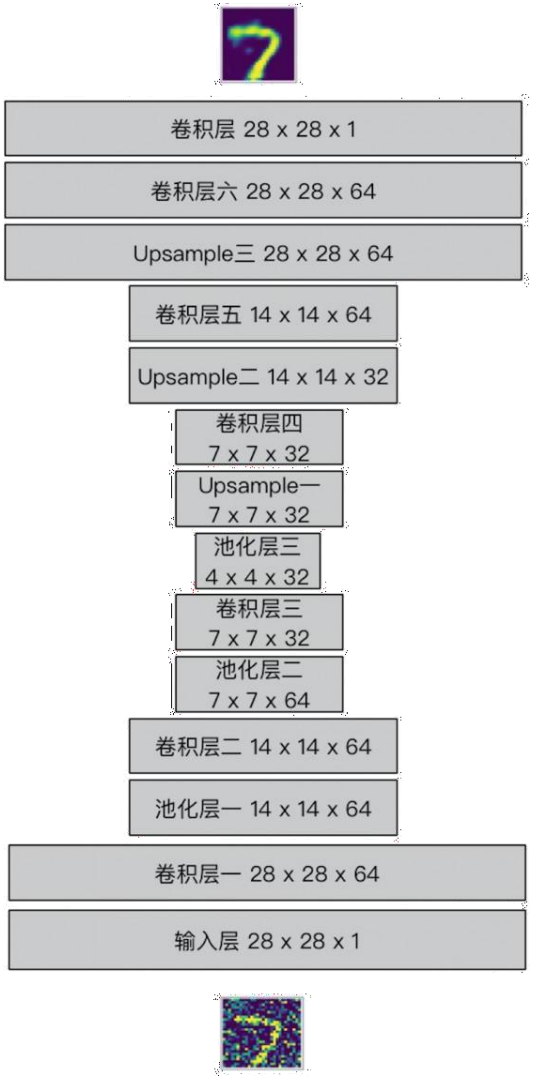
自编码器

- 堆叠自编码器 (stacked autoencoder, SAE)：
  - 将多个自编码器得到的隐层串联；
  - 所有层预训练完成后，进行基于监督学习的全网络微调



自编码器

- 数据输入(x_1, x_1)，$(x_2, x_2), \cdots, (x_N, x_N)$
- 期望回归函数为(x, x)，实际学到的回归函数$(x, F(x))$
- 其　中 $F(x) = S\left(W^{(2r)}(x)^{(2r)}\right) = (x)^{(2r+1)}$
  （r层编码，r层解码）
- 一般编码层每层结点数递减，解码层每层的节点数在递增（可用于降维）
- 自编码器是一种网络结构，可配合其他结构搭建深度网络（如卷积、池化等）

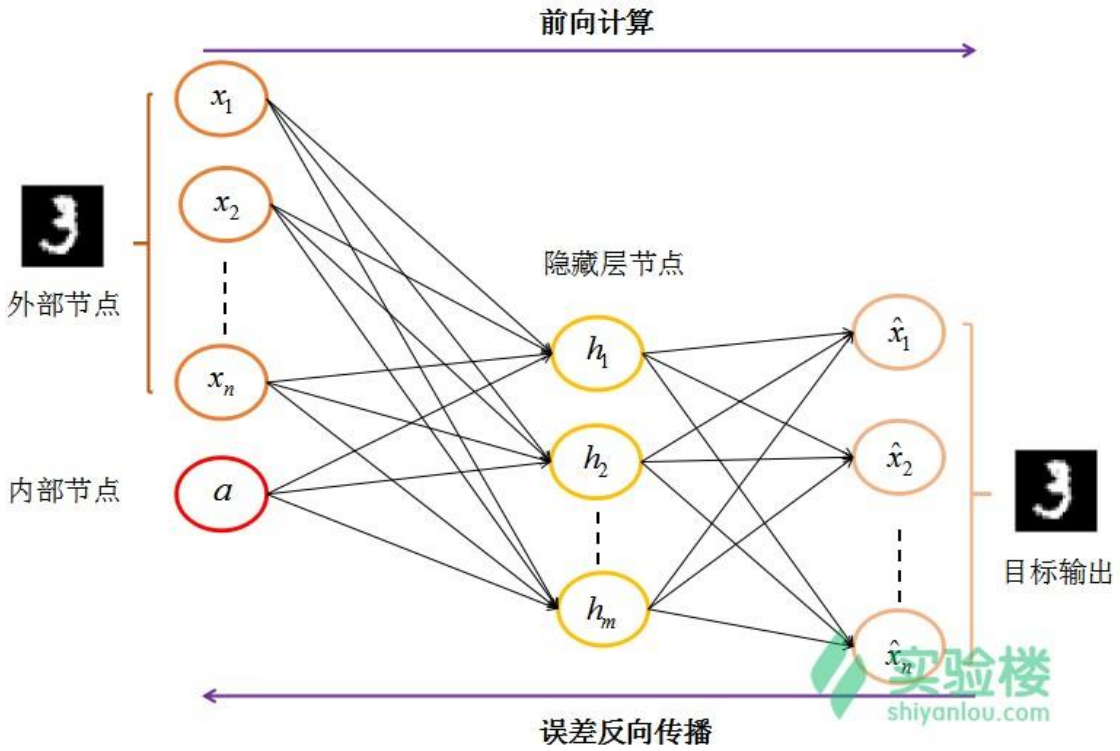


自编码器

- 根据经验风险最小化策略，应最小化的目标函数为

$$|I - F| = \sum_{k=1}^{N} \|x_k - F(x_k)\|^2$$

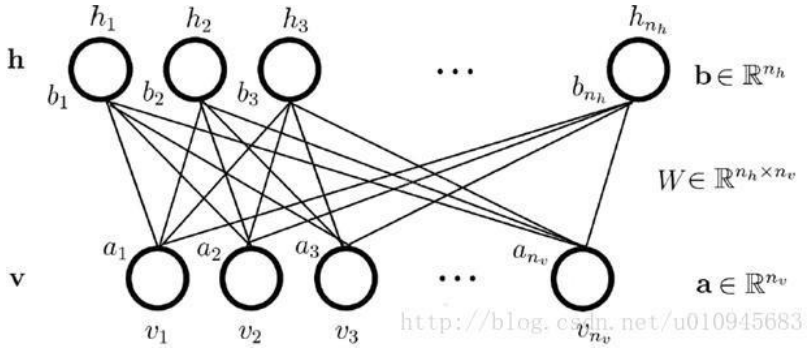
- 可以使用BP算法求解



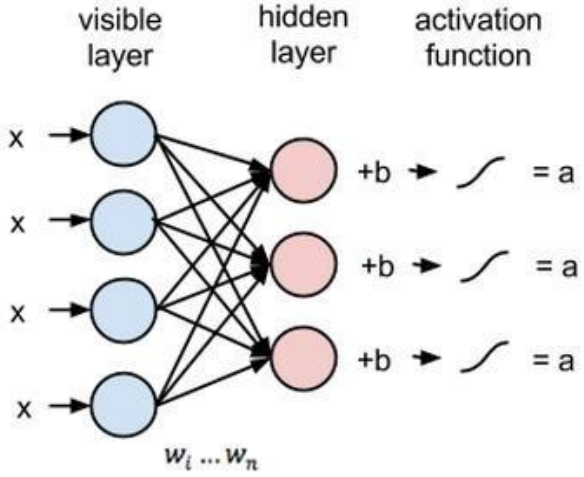
受限玻耳兹曼机 (RBM)

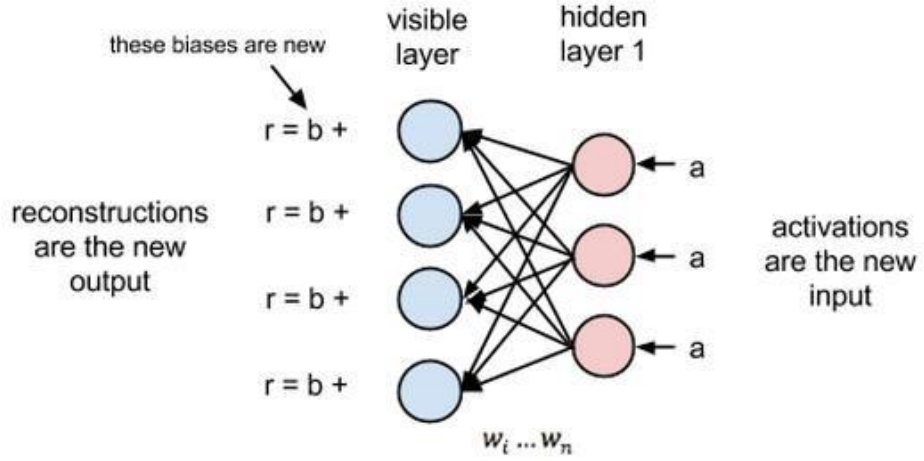
模型结构

- RBM是两层神经网络，包含可见层v（输入层）和隐藏层h
- 不同层之间全连接，
  层内无连接 → 二分图



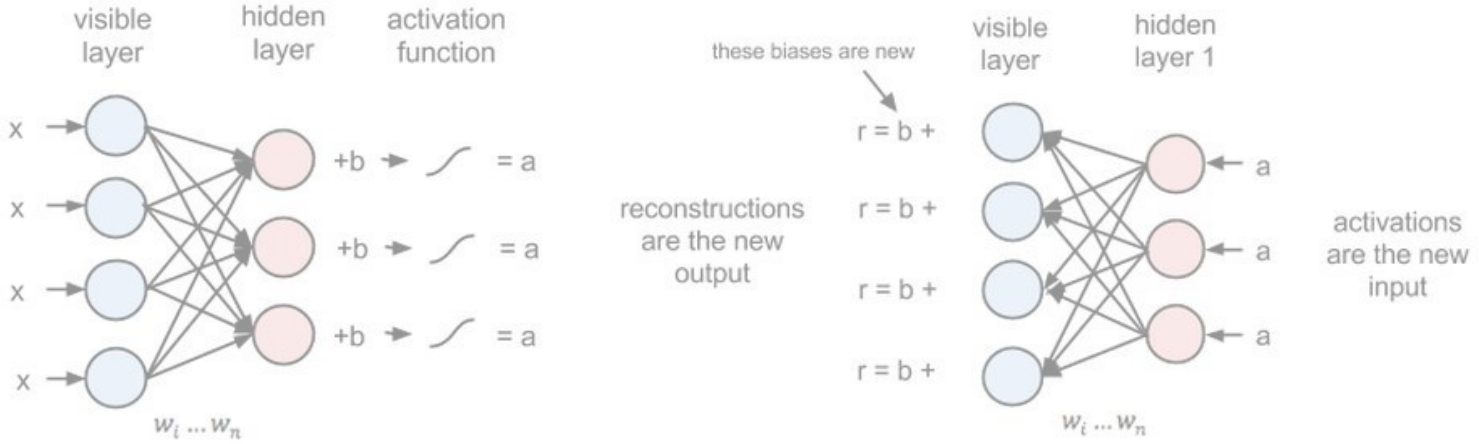
- 与感知器不同，RBM没有显式的重构过程：
  - ✓ 输入v，通过p(h|v)得到隐藏层h；输入h，通过p(v|h)得到v



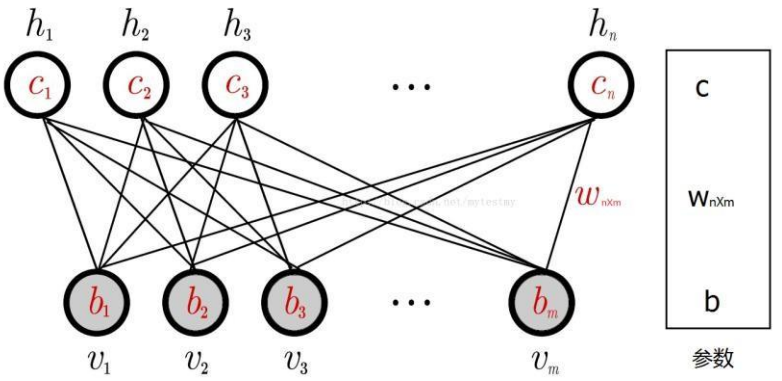


受限玻耳兹曼机 (RBM)

- 与感知器不同，RBM不区分前向和反向：
  - ✓ 输入v，通过p(h|v)得到隐藏层h；输入h，通过p(v|h)得到v



- 目的是让隐藏层得到的可见层v'与原来的可见层v分布一致，从而使隐藏层作为可见层输入的特征
- 两个方向权重w共享，偏置不同
- 模型参数：w, c, b

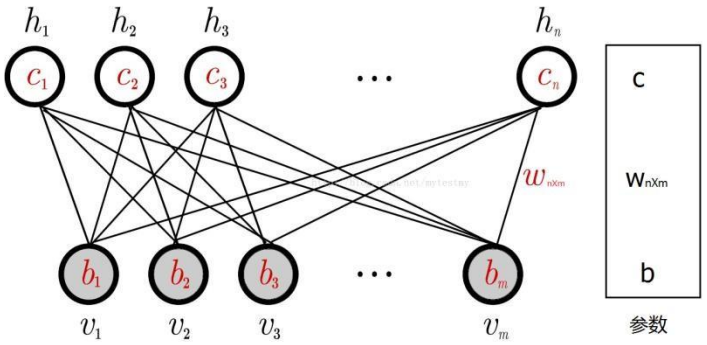


受限玻耳兹曼机 (RBM)

- **条件概率建模**

$$P(h_i = 1|v) = sigm(c_i + W_i v)$$

$$P(v_j = 1|h) = sigm(b_j + W'_j h)$$



- 联合概率→条件概率

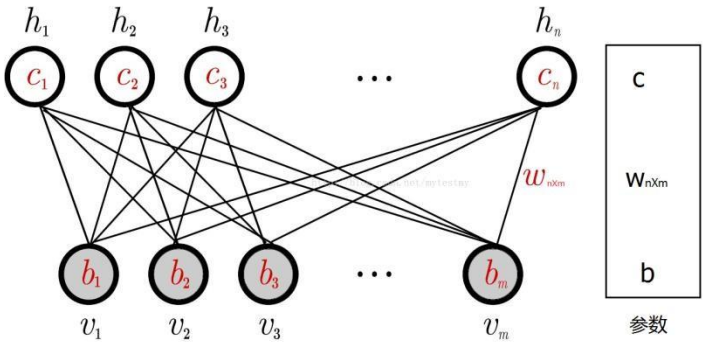
$$p(h|v) = \frac{p(h,v)}{p(v)}, p(v|h) = \frac{p(h,v)}{p(h)}$$

受限玻耳兹曼机 (RBM)

条件概率建模

$$P(h_i = 1|v) = sigm(c_i + W_i v)$$

$$P(v_j = 1|h) = sigm(b_j + W'_j h)$$

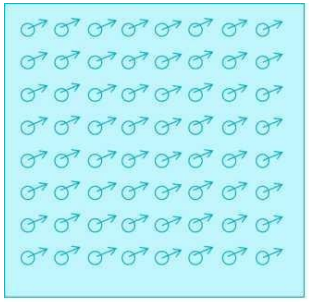


- 联合概率→条件概率

$$p(h|v) = \frac{p(h,v)}{p(v)}, p(v|h) = \frac{p(h,v)}{p(h)}$$

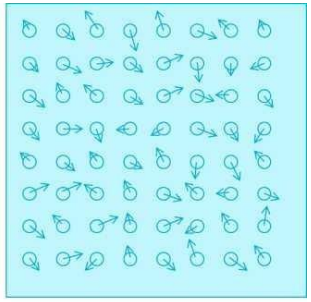
- 能量→联合概率
  - 统计物理学：能量越低越稳定，概率越大

$$p(x) = \frac{e^{-E(x)}}{Z}. \qquad Z = \sum_x e^{-E(x)}$$



有序，发生的概率小，熵小

能量高



无序，发生的概率大，熵大

能量低

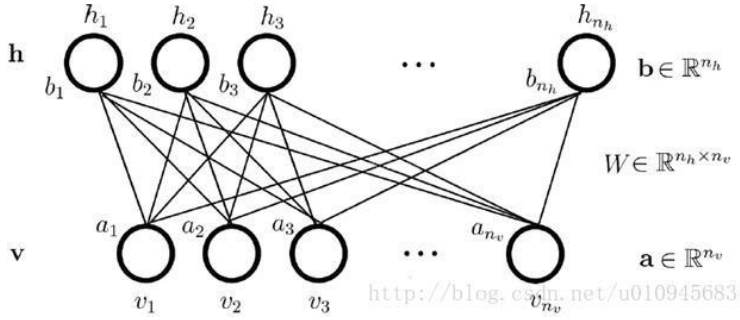
  - 能量的玻尔兹曼分布(Ising模型特殊形式)

$$E(v,h) = -b'v - c'h - h'Wv \qquad P(v,h) = \frac{1}{Z}e^{-E(v,h)}$$

受限玻耳兹曼机 (RBM)

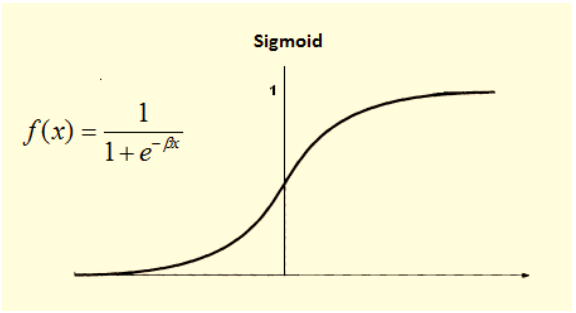
- **玻尔兹曼分布→sigmoid激活函数**

$$
\begin{aligned}
P(h|v) &= \frac{P(h,v)}{P(v)} \\
&= \frac{1}{P(v)}\frac{1}{Z}exp\{a^Tv + b^Th + h^TWv\} \\
&= \frac{1}{Z'}exp\{b^Th + h^TWv\} \\
&= \frac{1}{Z'}exp\{\sum_{j=1}^{n_h}(b_j^Th_j + h_j^TW_{j,:}v)\} \\
&= \frac{1}{Z'}\prod_{j=1}^{n_h}exp\{b_j^Th_j + h_j^TW_{j,:}v\}
\end{aligned}
$$



- 假设是标准玻尔兹曼分布: 所有节点是二进制变量(0,1)

$$
\begin{aligned}
P(h_j = 1|v) &= \frac{P(h_j = 1|v)}{P(h_j = 1|v) + P(h_j = 0|v)} \\
&= \frac{exp\{b_j + W_{j,:}v\}}{exp\{0\} + exp\{b_j + W_{j,:}v\}} \\
&= \frac{1}{1 + exp\{-(b_j + W_{j,:}v)\}} \\
&= sigmoid(b_j + W_{j,:}v)
\end{aligned}
$$



Hinton 证明了这件事。有个插曲，ReLU提出后，Hinton不高兴

受限玻耳兹曼机 (RBM)

模型求解

- **优化目标**：网络表示的概率分布与输入样本分布尽可能接近
- 可见层v的似然：

$$p(v) = \sum_h p(h, v) = -b'v - \sum_i \log(1 + e^{(c_i + W_i v)})$$

- 最大似然的梯度下降求解：

$$-\frac{\partial \log p(v)}{\partial W_{ij}} = E_v[p(h_i|v) \cdot v_j] - v_j^{(i)} \cdot sigm(W_i \cdot v^{(i)} + c_i)$$

$$-\frac{\partial \log p(v)}{\partial c_i} = E_v[p(h_i|v)] - sigm(W_i \cdot v^{(i)})$$

$$-\frac{\partial \log p(v)}{\partial b_j} = E_v[p(v_j|h)] - v_j^{(i)}$$

- 梯度下降 → 基于Gibbs采样的对比散度：

$$Z = \sum_{v,h} e^{-E(v,h)}$$

状态组合数目太多$(2^{n_v+n_h})$，难以直接求和 → **采样来近似计算**

受限玻耳兹曼机 (RBM)

RBM到DBN（深度信念网络）

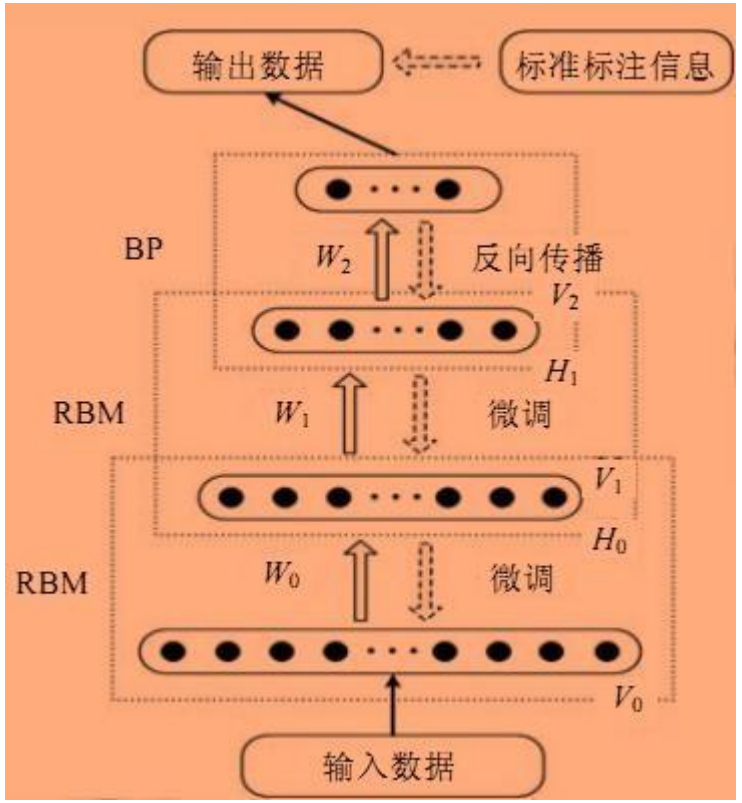
- 一个DBN模型由若干个RBM堆叠而成，最后加一个监督层(如BP 网络)
- 训练过程由低到高逐层训练：

  ①最底部RBM以原始输入数据训练

  ②将底部RBM抽取的特征作为顶部RBM的输入继续训练

  ③重复这个过程训练尽可能多的RBM层

  ④基于监督信息通过全局优化算法对网络进行微调，使模型收敛

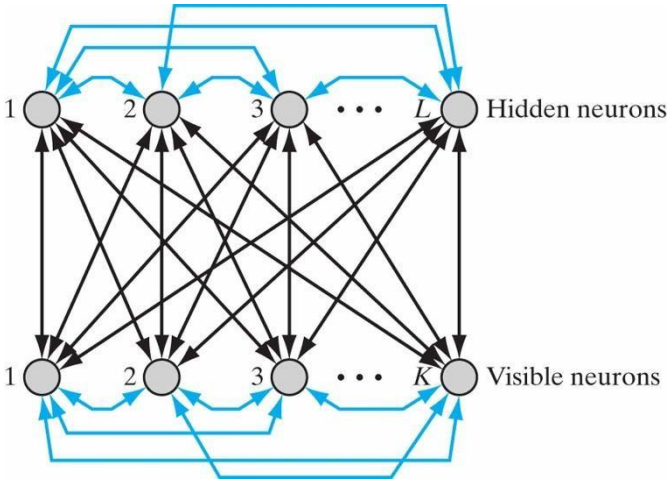


- DBN (Deep Belief Network) VS DBM (Deep Boltzmann Machine)：
  - ✓ DBM没有监督层，是若干个RBM的直接堆叠
  - ✓ 无向图模型，每两层间互有反馈

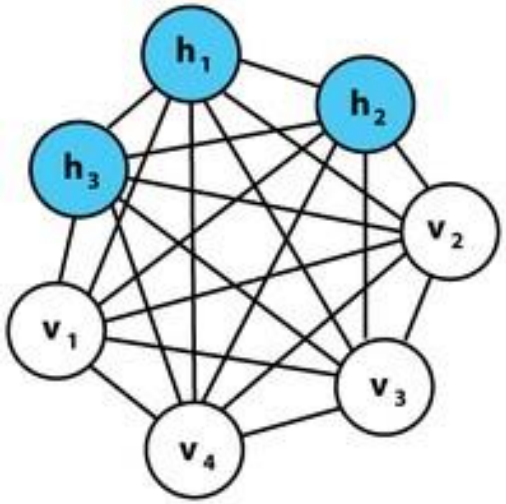
受限玻耳兹曼机 (RBM)

一般玻尔兹曼机(BM)

- 可见层和隐层内部结点之间可连接
- 具有很强大的无监督学习能力，能够学习数据中复杂的规则
- 随机神经网络和递归神经网络的一种，Hinton等1985年发明

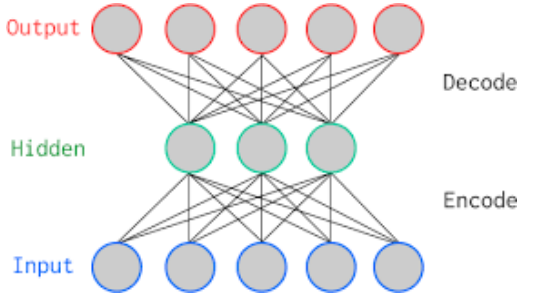


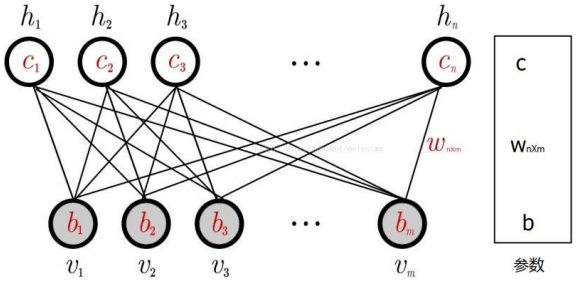
- 全连接图，复杂度很高
  - 难以准确计算BM所表示的分布
  - 难以抽样得到服从BM所表示分布的随机样本



自编码器 VS 受限玻耳兹曼机

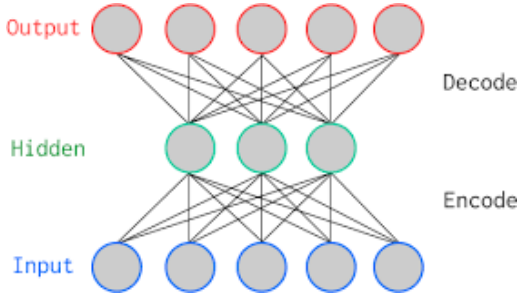
- **结构上：**
  - 自编码器编码和解码函数不同：W1, W2
  - RBM共享权重矩阵W，两个偏置向量

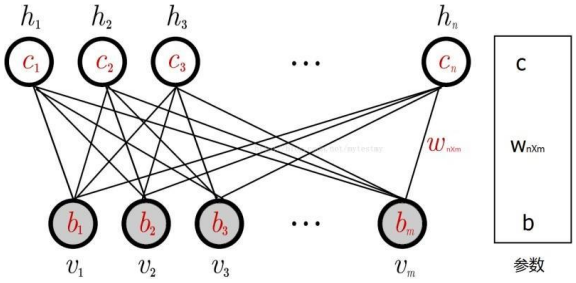




自编码器 VS 受限玻耳兹曼机

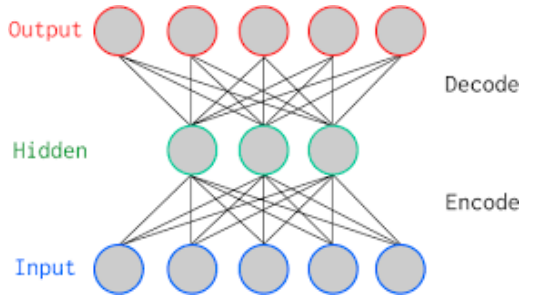
- **结构上：**
  - 自编码器编码和解码函数不同:W1, W2
  - RBM共享权重矩阵W，两个偏置向量
- **原理上：**
  - 自编码器通过非线性变换学习特征，是确定的，特征值可以为任何实数；
  - RBM基于概率分布定义，高层表示为底层特征的条件概率，输出只有两种状态（未激活激活），用二进制0/1表示；



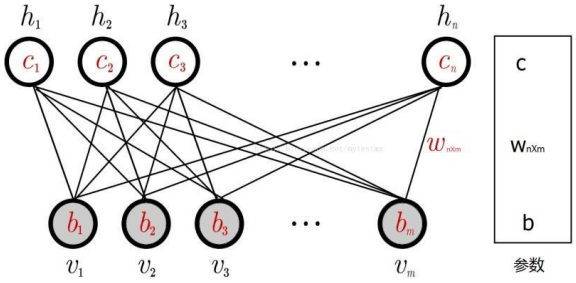


自编码器 VS 受限玻耳兹曼机

- **结构上：**
  - 自编码器编码和解码函数不同:W1, W2
  - RBM共享权重矩阵W，两个偏置向量



- **原理上：**
  - 自编码器通过非线性变换学习特征，是确定的，特征值可以为任何实数；
  - RBM基于概率分布定义，高层表示为底层特征的条件概率，输出只有两种状态（未激活激活），用二进制0/1表示；

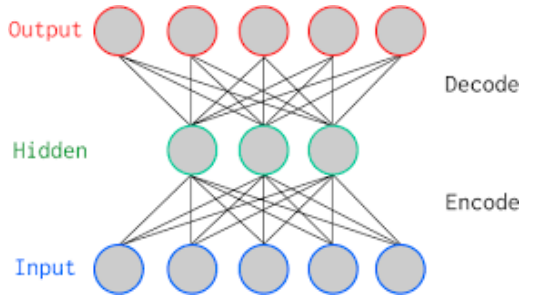


- **训练优化：**
  - 自编码器通过最损失函数L最小化重构输入数据，直接用BP优化求解
  - RBM基于最大似然，能量函数偏导无法直接计算，基于采样方法进行估计

自编码器 VS 受限玻耳兹曼机

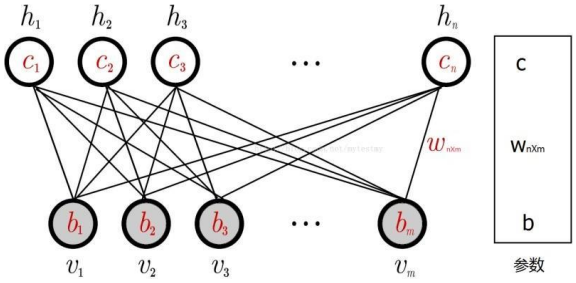
结构上：

- 自编码器编码和解码函数不同:W1, W2
- RBM共享权重矩阵W，两个偏置向量



原理上：

- 自编码器通过非线性变换学习特征，是确定的，特征值可以为任何实数；
- RBM基于概率分布定义，高层表示为底层特征的条件概率，输出只有两种状态（未激活激活），用二进制0/1表示；



训练优化：

- 自编码器通过最损失函数L最小化重构输入数据，直接用BP优化求解
- RBM基于最大似然，能量函数偏导无法直接计算，基于采样方法进行估计

生成/判别模型：

- RBM对联合概率密度建模，是生成式模型；
- 自编码器直接对条件概率建模，是判别式模型。

预训练的实际作用

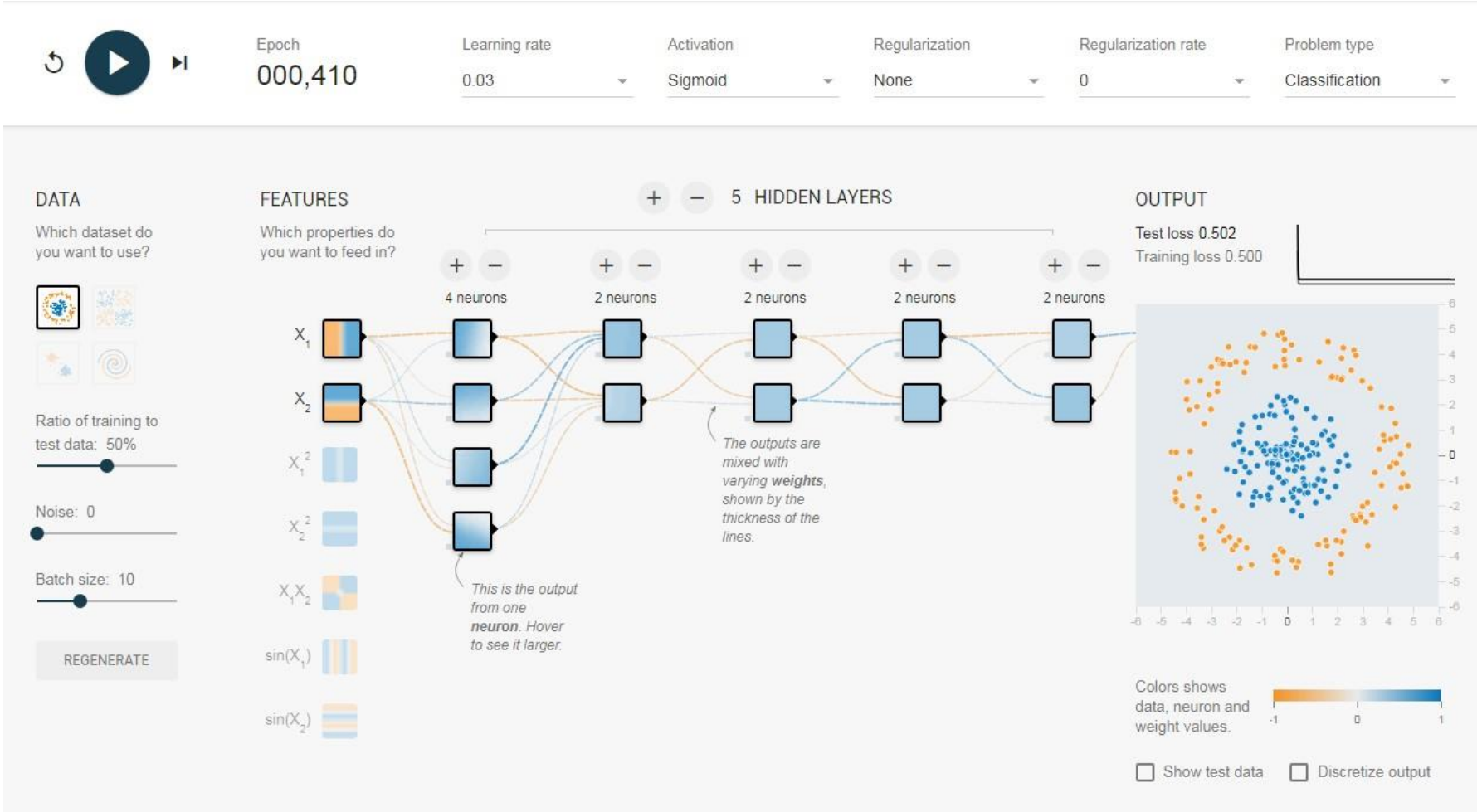
- **初衷用于无监督逐层预训练**
  - 当时的历史条件（没有新激活函数、没有dropout等优化技术）下，逐层无监督预训练使得深度网络的训练有了可能（2006-2012）
  - 但逐层预训练无法本质上解决梯度消失等问题
  - 新的激活函数 + 优化方法 + 更大量的标注训练数据 → 预训练很少使用（ 2012- ）

预训练的实际作用

- **初衷用于无监督逐层预训练**
  - 当时的历史条件（没有新激活函数、没有dropout等优化技术）下，逐层无监督预训练使得深度网络的训练有了可能（2006-2012）
  - 但逐层预训练无法本质上解决梯度消失等问题
  - 新的激活函数 + 优化方法 + 更大量的标注训练数据 → 预训练很少使用（ 2012- ）
- **DNN VS DBN**
  - DNN是前馈神经网络，训练方法是BP
  - 隐层激活函数使用ReLU → 改善梯度消失
  - 输出层激活函数是softmax，目标函数是交叉熵+大量标注数据→避免差的局部极小值
  - 正则化+dropout → 改善过拟合
  - **没有使用逐层预训练**

深层神经网络的问题：梯度消失

http://playground.tensorflow.org/



解决梯度消失

- **Layer-wise Pre-train**: 2006年Hinton等提出的训练深度网路的方法，用无监督数据作分层预训练，再用有监督数据fine-tune

DBN, 2006

- **ReLU**: 新的激活函数解析性质更好，克服了sigmoid函数和tanh函数的梯度消失问题

AlexNet, 2012

- **辅助损失函数**：e.g. GoogLeNet中的两个辅助损失函数，对浅层神经元直接传递梯度

Inception V1, 2013

- **Batch Normalization**: 逐层的尺度归一

Inception V2, 2014

- **LSTM**: 通过选择记忆和遗忘机制克服RNN的梯度消失问题，从而可以建模长时序列

Inception

玻尔兹曼机的理论和应用意义

- BM和RBM数学上很漂亮,且有统计物理学支撑;
- 但受结构限制严重，生成式模型效果往往不如判别式模型;
- 主流的深度学习平台甚至都不支持RBM和预训练

吴恩达用1.6万块CPU构建了全球最大的电子模拟神经网络，并向其展示自YouTube上随机选取的1000万段视频，考察其能够学到什么

结果显示，在无外界指令的自发条件下，该人工神经网络自主学会了识别猫的面孔

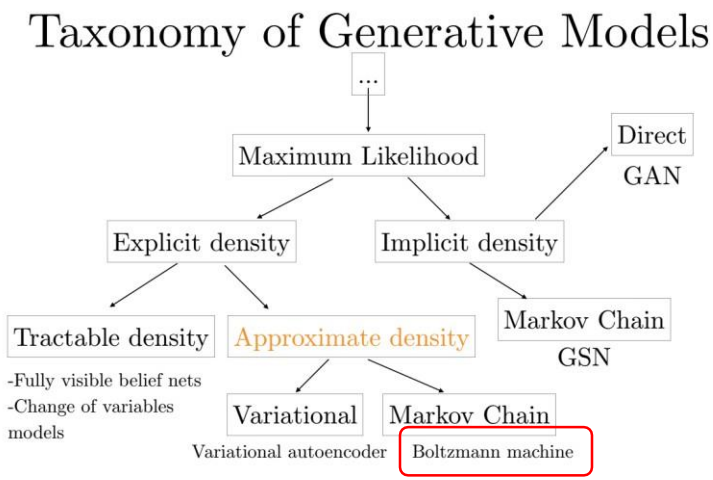
玻尔兹曼机的理论和应用意义

- BM和RBM数学上很漂亮,且有统计物理学支撑;
- 但受结构限制严重，生成式模型效果往往不如判别式模型;
- 主流的深度学习平台甚至都不支持RBM和预训练

- **理论：**
  - **模型结构**：网络拓扑结构优化
  - **学习算法**：非线性优化过程近似
- **应用：**

作为一种概率生成式模型应用到了协同滤波推荐、数据降维、时间序列降维问题。

[Gaussian-bernoulli deep boltzmann machine]

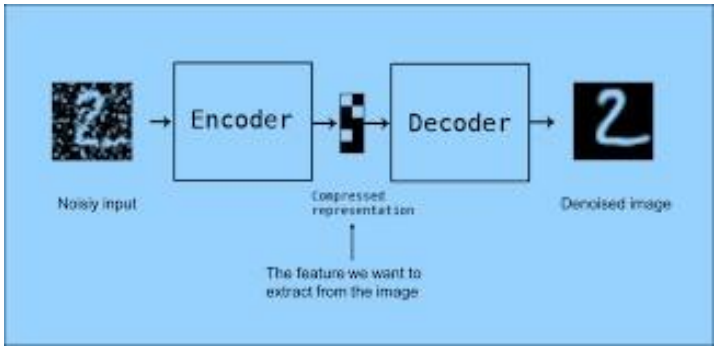


自编码器变种

正则自编码器 （Regularized AE）

- 应用：使提取的特征表达符合某种性质
- 惩罚大权重的L2正则化：

$$\mathcal{J}_{AE+wd}(\theta) = \sum_{\mathbf{x}\in S} L(\mathbf{x}, g(f(\mathbf{x}))) + \lambda \sum_{i,j} W_{i,j}^2,$$



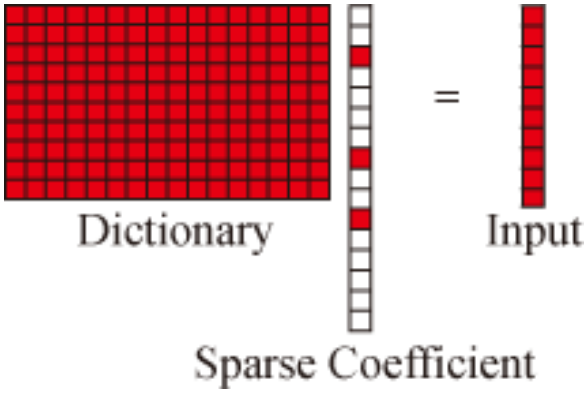
稀疏自编码器 （Sparse AE）

- 应用：提取稀疏特征表达

（**假设**：高维而稀疏的表达是好）

- 神经元平均激活度：

$$\hat{\rho}_j = \frac{1}{N} \sum_{i=1}^{N} h_j(\mathbf{x}^{(i)}),$$



- 稀疏约束：限制神经元平均激活度在一个很小的值

$$\mathcal{J}_{AE+wd+sp}(\theta) = \sum_{\mathbf{x}\in S} L(\mathbf{x}, g(f(\mathbf{x}))) + \lambda \sum_{i,j} W_{i,j}^2 + \beta \sum_{j=1}^{m} KL(\rho||\hat{\rho}_j).$$

K-L散度，

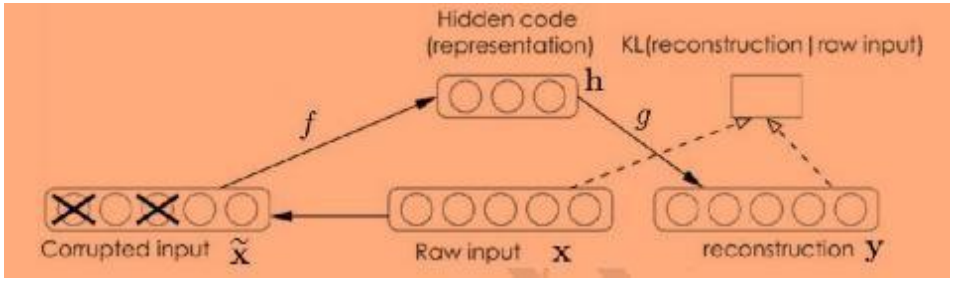
衡量两个分布的相异性

自编码器变种

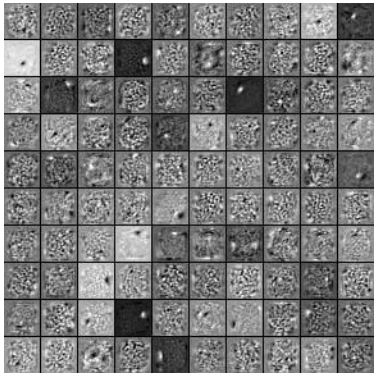
去噪自编码器 (Denoising AE)

- 应用：提取鲁棒特征表达（假设：能够对“被污染/破坏”的原始数据编码、解码，还能恢复真正的原始数据，这样的特征才是好的）
- 污染：$\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2 I).$
- 编解码：$\mathbf{h} = f(\tilde{\mathbf{x}}) := s_f(W\tilde{\mathbf{x}} + \mathbf{p});$

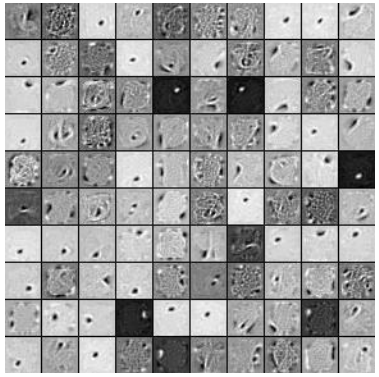
$$\mathbf{y} = g(\mathbf{h}) := s_g(\widetilde{W}\mathbf{h} + \mathbf{q}),$$



- 学习的特征表达示例



无噪声

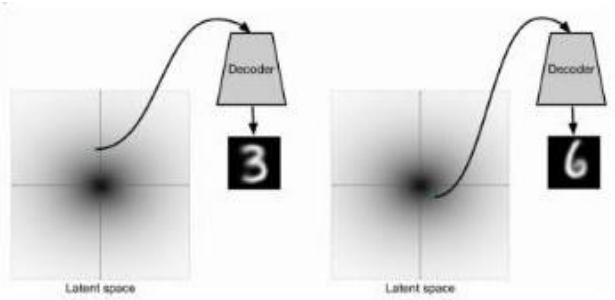


加入白噪声

自编码器变种

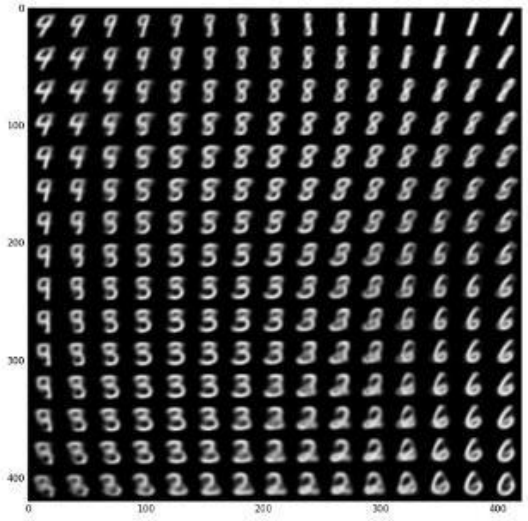
变分自编码器 （Variational AE）

- **原理**：基于隐层特征表达空间Z，通过解码层，生成样本
- **非监督生成模型**，与对抗式生成网络GAN关系密切，是深度学习-概率图模型桥梁
- 应用：



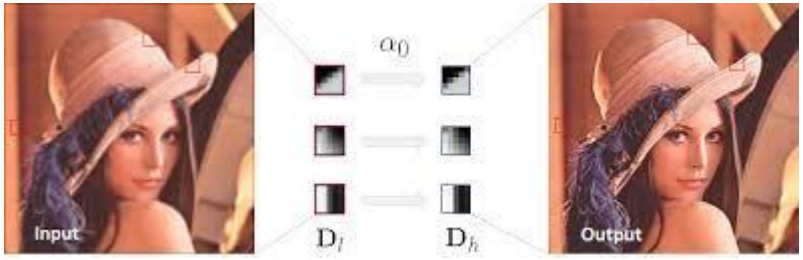
隐层表达空间Z

✓ 数据生成



✓ 缺失数据填补

✓ 图像超分辨率



变分自编码器demo

http://vdumoulin.github.io/morphing_faces/online_demo.html

人脸图像

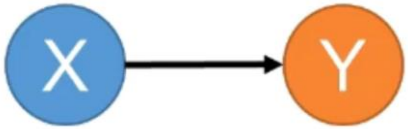
Random

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	

变分自编码器demo

@邱锡鹏

因果推断



很多时候，模式识别要解决的是一个相关性的问题

相关性不可靠：Yule-Simpson 悖论

Population	Survive	Die	Survive Rate
Treatment	20	20	50%
Control	16	24	40%

X: 处理与否

Y：存活率

结论：X与Y正相关

Male	Survive	Die	Survive Rate
Treatment	18	12	60%
Control	7	3	70%
Female	**Survive**	**Die**	**Survive Rate**
Treatment	2	8	20%
Control	9	21	30%

Z: 性别

结论：X与Y负相关

相关关系可能由于context等混杂因素而改变

这也说明，相关性是相当脆弱的，很容易改变

相关性 VS 因果性

统计结果发现：同时患肺炎和哮喘的患者死于肺炎的概率要低于无哮喘患者

→基于相关性，哮喘可能成为肺炎患者的保护因素

有哮喘史的肺炎病人被直接送入ICU重症病房

↓

降低同时患肺炎和哮喘患者的死亡率

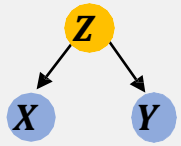
因果性=相关性+忽略的因素

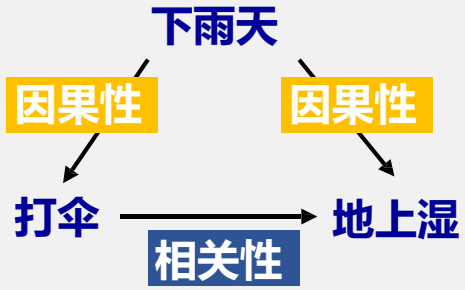
因果性

排除其他因素后，Y仅由X影响

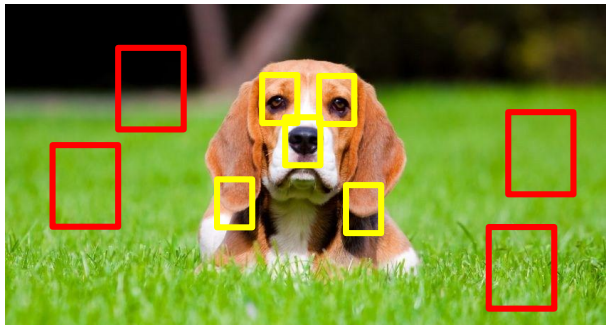
相关性

Y的发生往往伴随着X





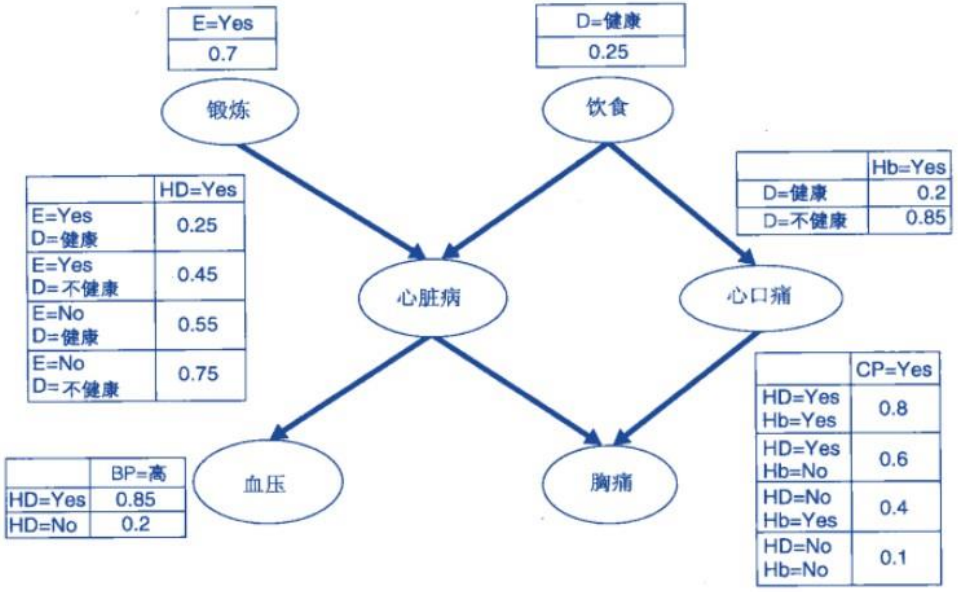
相关性 VS 因果性



A **man** sitting at a desk with a laptop computer.

联结主义 VS 贝叶斯：相关性 VS 因果性

$$P(\text{心脏病}=\text{No}|\text{锻炼}=\text{No},\ \text{饮食}=\text{健康})$$
$$= 1 - P(\text{心脏病}=\text{Yes}|\text{锻炼}=\text{No},\ \text{饮食}=\text{健康})$$
$$= 1 - 0.55 = 0.45$$

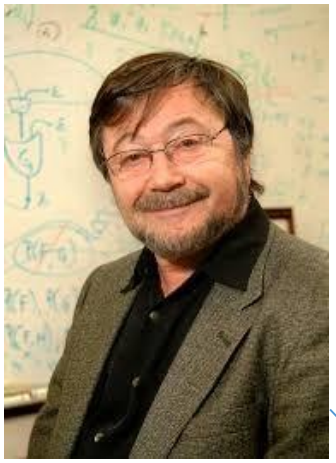


Judea Pearl

Michael Jordan

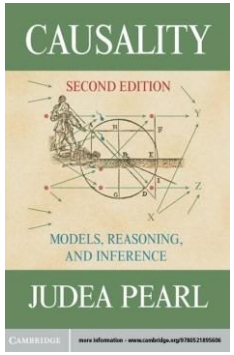
联结主义 VS 贝叶斯：相关性 VS 因果性

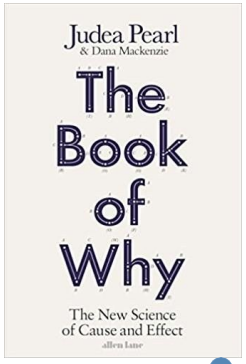
【NIPS2017】深度学习真的不需要理论指导了？图灵奖得主讲座无人问津，贝叶斯之父Judea Pearl落寞身影背后引人深思



Judea Pearl

因果网络：基于贝叶斯网络增加因果限制-父节点必须是子节点的原因





联结主义 VS 贝叶斯：相关性 VS 因果性

Michael Jordan

概率图模型专家，NIPS学 阀， 被 Semantic Scholar钦定为CS领域最具影响力学者

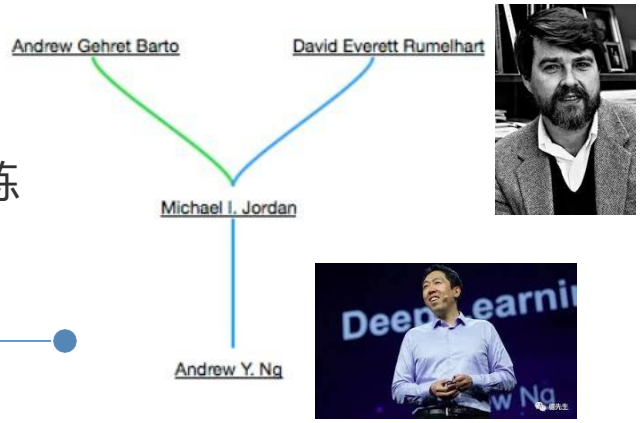
Science Home News Journals Topics Careers

Who's the Michael Jordan of computer science? New tool ranks researchers' influence

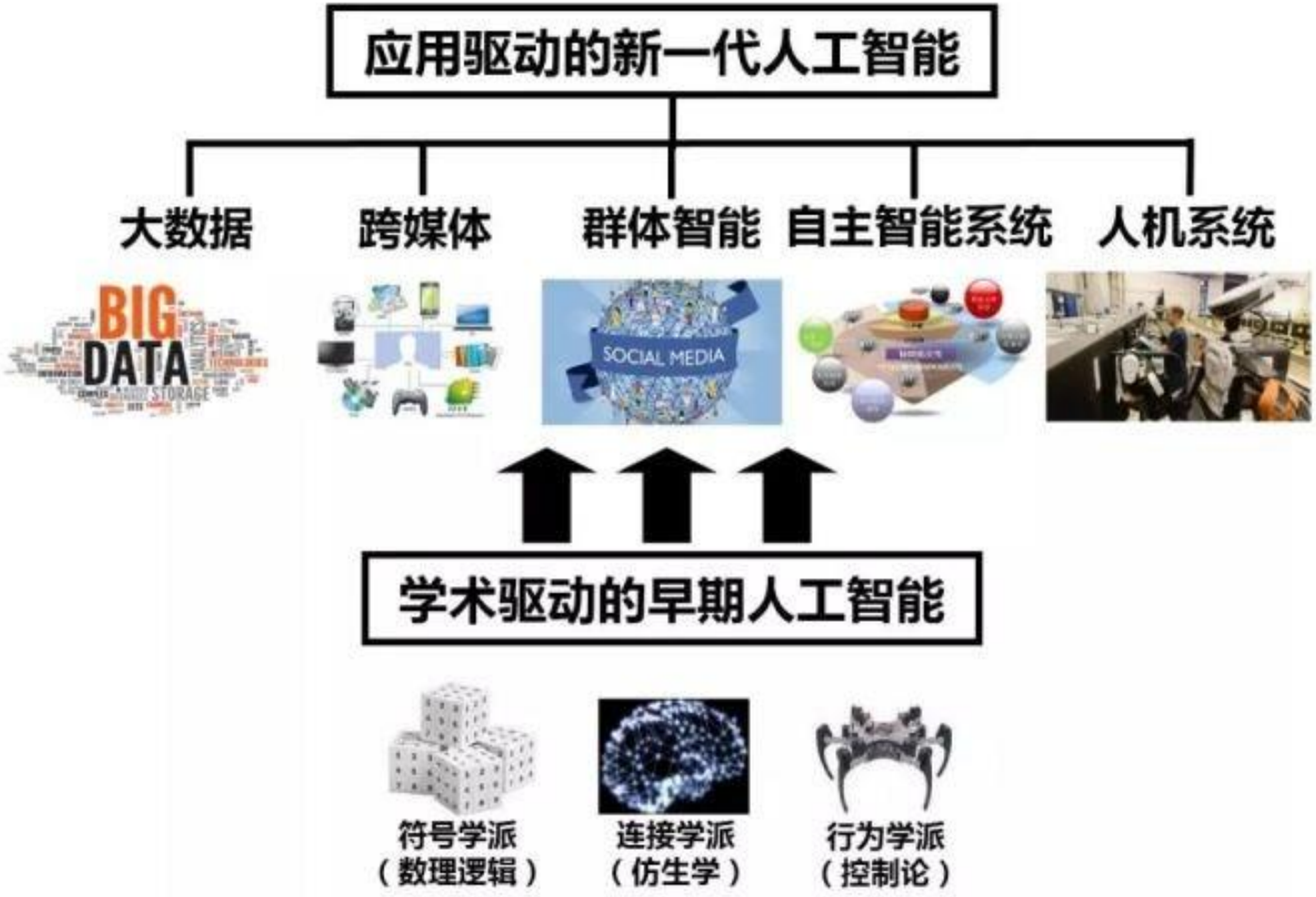
Top 50 authors in computer science

Author	Influencial	Citation	S2 Citations
Michael I. Jordan	1185	5745	24452

Rumelhart:
BP算法引入神经网络训练
Jordan: RNN提出者
吴恩达：GPU加速训练



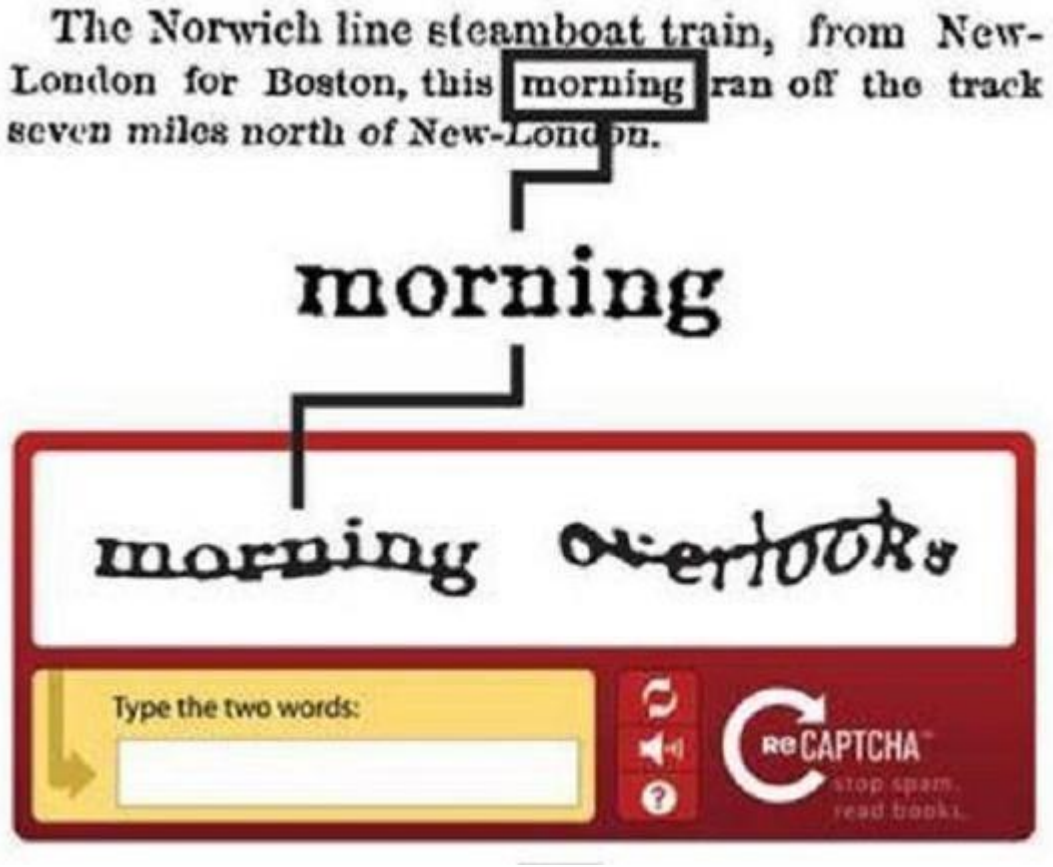
新一代人工智能主要发展领域



资料来源：《新一代人工智能发展白皮书》

群体智能

群体智能：再看验证码


The Norwich line steamboat train, from New-London for Boston, this morning ran off the track seven miles north of New-London.
morning
morning
Type the two words:
reCAPTCHA
stop spam.
read books.

验证码：师夷长技以制夷

ESP：寓“教”于乐

图像标注游戏

Player 1

guess: BOAT

guess: WATER

guess: RIVER

Score! Agreement on ‘BOAT’.

Player 2

guess: PORT

guess: BOAT

Score! Agreement on ‘BOAT’.

升级版：Peekaboom

Peekaboom: Boom gets an image along with a word related to it, and must reveal parts of the image for Peek to guess the correct word. Peek can enter multiple guesses that Boomcan see.

图像分割游戏

更多例子

Luis Von Ahn

Learning a language while translating the web

reCAPTCHA

WIKIPEDIA
The Free Encyclopedia

amazon
mechanical turk
Artificial Artificial Intelligence

waze

高德地图
amap.com

THANKS