DL

深度学习介绍

赵亚伟

中国科学院大学 大数据分析技术实验室

2018.6.23

- 现状
- 神经网络
- 深度学习
  - 介绍
  - 常见模型
    - Stacked Auto-Encoder
    - Convolutional Neural Network
    - Deep Belief Network
- 杂项

现状

- 2006年7月，机器学习界的著名学者Geoffery Hinton 和他的学生Ruslan Salakhutdinov在《Science》上发表了一篇关于深层神经网络训练算法的文章，引起了深度学习在学术界和工业界的关注；
- 2012年5月到10月，为了回应对于深度学习方法的质疑，也为了测试深度学习算法的真实效果，Geoffery Hinton和他的另外一个学生Alex Krizhevsky参加了ImageNet大规模视觉识别竞赛，以大幅领先优势取得第一名；
- 2012年6月，机器学习和人工智能的知名学者Andrew Ng等人与谷歌经过多年的合作，在语音识别与图像目标识别方面取得了突破性的进展；
- 2012年11月，在二十一世纪计算大会上，微软首席研究官Rick Rashid展示了一套基于深层神经网络模型的语音识别系统，能实时流畅翻译，切显著地降低了错误率，这在语音识别领域是一个突破；
- 2013年1月，百度创始人兼CEO李彦宏在年会上宣布将成立百度研究院；
- 2013年7月，谷歌收购了深度学习初创公司DNNResearch Inc.；

现状

- 2013年10月，雅虎收购图像识别公司LookFlow，正式加入深度学习研究队伍；
- 2015到2016年，苹果收购人工智能研究公司VocalIQ，Percepti，Emotient，Turi等，强化Siri和摄像等应用上的优势；
- 2015年11月，谷歌发布人工智能系统TensorFlow并宣布开放源代码；
- 2016年5月，亚马逊发布了一个开源的使用GPU训练和部署深层神经网络开源工具DSSTNE；
- 2016年8月，英特尔开发者大会上，英特尔宣布将于2017年推出专门为人工智能和深度学习而设计的最新一代CPU—Intel Xeon Phi处理器，代号Knights Mill，英特尔宣称其运算能力比对手NVIDIA的Kepler系列GPU产品快两倍以上。
- 2016年10月，NVIDIA 发布了新版本的通用并行计算架构库：统一计算设备架构（Compute Unified Device Architecture，CUDA）8.0，以及深度学习专用GPU 加速库：cuDNN 5.0；
- 2016年11月，在2016全球超级计算机大会（SC16）上，AMD 宣布推出新版Radeon开放计算平台（Radeon Open Compute Platform，ROCm），以及用于GPU 加速器的免费开源库MIOpen。

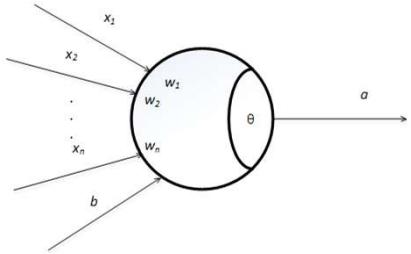
- 现状
- **神经网络**
- 深度学习
  - 介绍
  - 常见模型
    - Stacked Auto-Encoder
    - Convolutional Neural Network
    - Deep Belief Network
- 杂项

神经网络

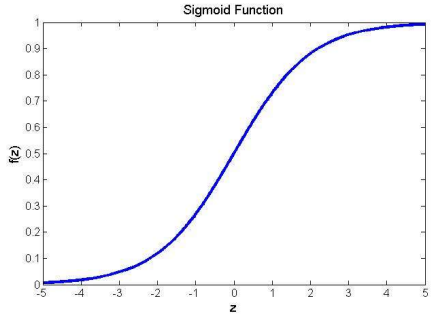
- 在机器学习与认知识别领域中，人工神经网络是一类模拟生物神经网络的模型，基于大量训练数据，用来预测（决策问题）或估计目标函数模型。
- 人工神经网络一般呈现为相互关联的“神经元”之间相互交换信息的系统。
- 在神经元的连接中包含有可以根据训练样本调整的权重，使得神经网络可以自适应输入样本，并且拥有学习能力。
- 作为机器学习方法的一种，神经网络算法可以用来处理一系列传统机器方法无法处理，或者处理难度较大的问题，包括计算机视觉、语音识别等任务。

基本结构

- 神经网络的基本单元是神经元。通过对所有输入进行加权求和，之后进行非线性映射得到该神经元的输出值。



神经元模型



$$f(x)=\frac{1}{1+e^{-x}}$$

常用激活函数：

ReLU函数

S型函数

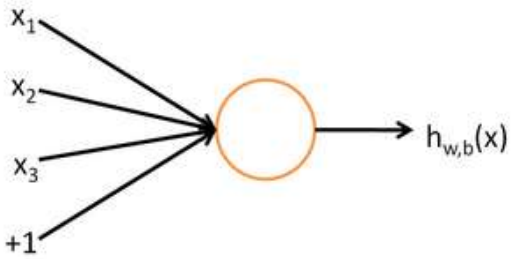
双曲正切函数

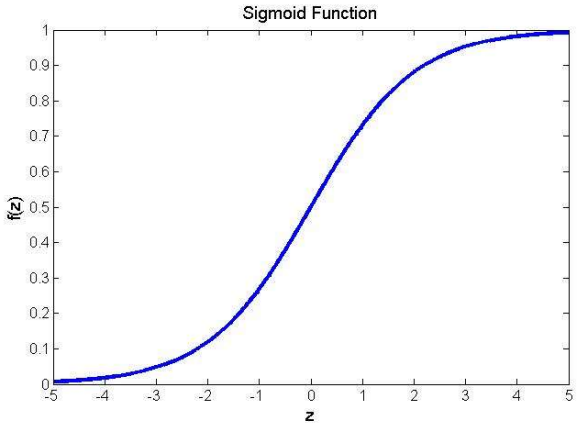
- **神经网络按照拓扑结构，大体分为层状与网状两大类。**

神经网络

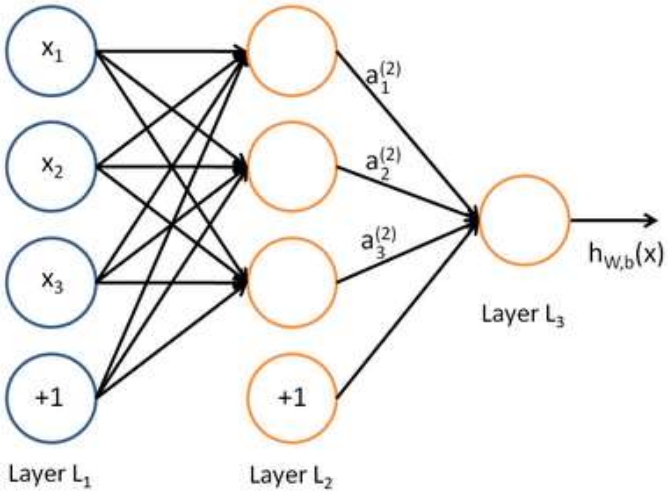
输出： $h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$

激活函数： $f : \Re \mapsto \Re$





神经网络



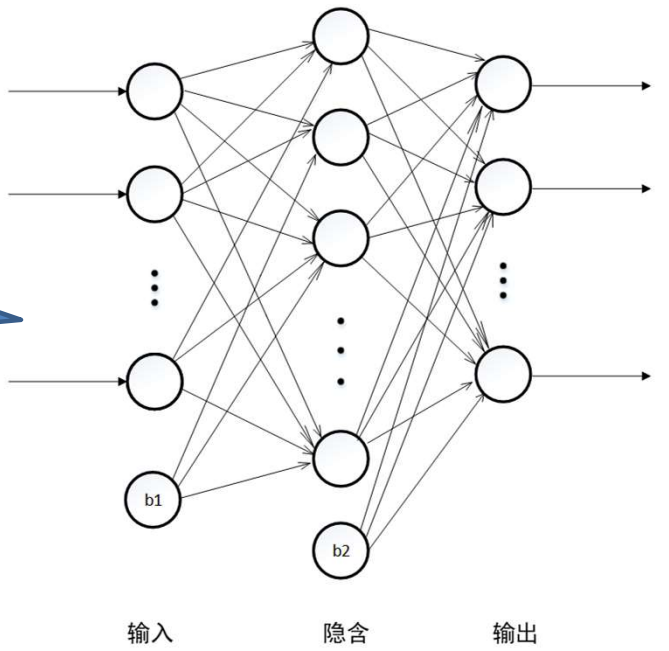
$$
\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

BP网络(一种常见的前馈网络)

- 前馈网络的逐层计算：
  - 输入值从输入层神经元通过加权连接逐层前向传播，经过隐含层，最后到达输出层得到输出。在信号的前向传播过程中，网络的权值是固定不变的，每一层神经元的状态只影响下一层神经元的状态（对下一层有指导），各层间没有反馈。
- 前馈与BP是两个概念
- 反向传播（Back Propagation）算法：
  - 网络的实际输出与期望输出之间的差值即为误差信号。误差信号由输出端开始逐层反向传播。在误差信号反向传播的过程中，网络的权值根据误差的梯度进行调节，通过权值的不断修正使网络的实际输出更加接近期望输出。

前馈：基于已知的类别进行网络训练即为前馈，前馈即有指导的

与前馈相反的是反馈：无监督学习，通过反馈逐步达到一个平衡状态。
反馈网络又称递归网络



前馈网络结构

说明

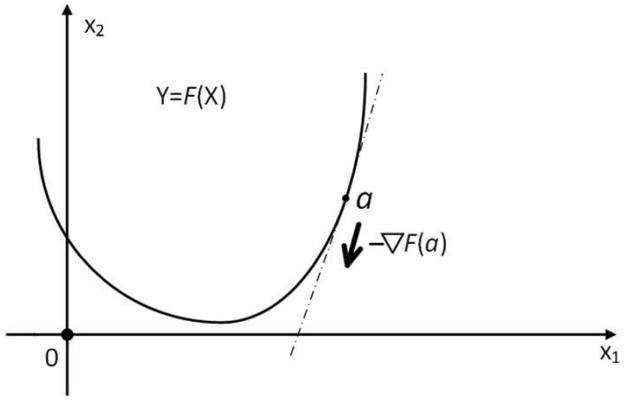
- 代价函数

在遇到回归问题时，指定代价函数 $J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$ 变量的真实值和预测值的距离最小；

代价函数描述了网络输出与真实值之间的误差；

网络模型的训练过程即：通过随机梯度下降的方法最小化代价函数以提高网络精度；

可以在代价函数中引入其他约束以满足设定要求。



反向传播算法

反向传播算法可表示为以下几个步骤：

1. 进行前馈传导计算，利用前向传导公式，得到 $L_2, L_3, \ldots$ 直到输出层L_{n_l}的激活值。
2. 对输出层（第n_l层），计算：

$$\delta^{(n_l)} = -(y - a^{(n_l)}) \bullet f'(z^{(n_l)})$$

3. 对于$l = n_l - 1, n_l - 2, n_l - 3, \ldots, 2$ 的各层，计算：

$$\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \bullet f'(z^{(l)})$$

4. 计算最终需要的偏导数值：

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^T,$$

$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)}.$$

5. 根据残差对参数W和b做出更新：

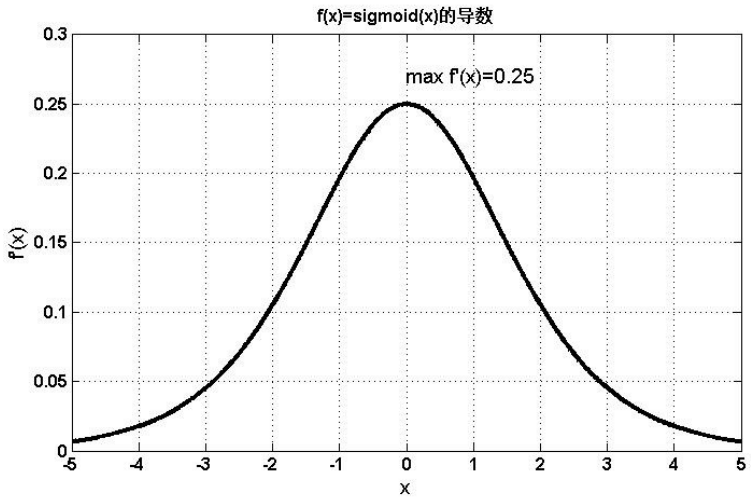
$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b)$$

反向传播：是指误差是输出端与已知的类别进行计算逐层向前传播，逐层调整权值的过程

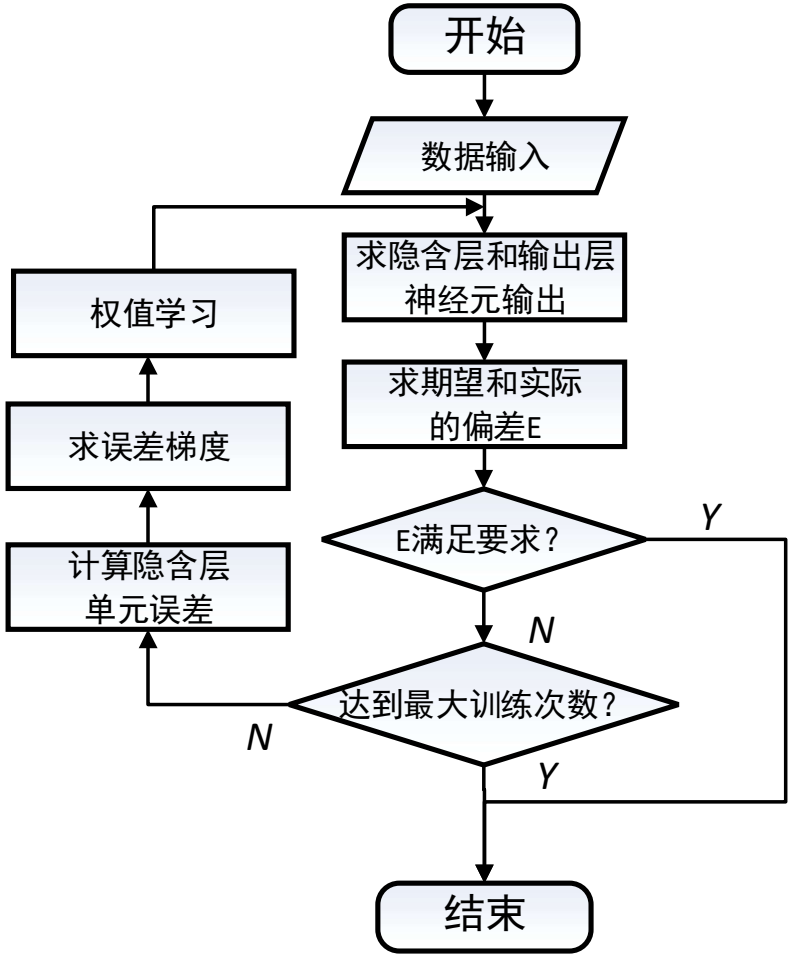
正向传播是根据输入，经过权值、激活函数、隐藏层，最终获得输出

反向传播与梯度下降



S型函数导数

$$f(z)=\frac{1}{1+\exp(-z)} \qquad f'(z)=f(z)(1-f(z))$$



BP算法流程

主要问题

主要问题

训练过程易陷入局部极小值，从而得不到全局最优解；

计算量大，训练次数多，使得学习效率低，收敛速度慢；

对于隐含层个数和隐含层神经元节点的个数选择，至今还没有一个具体的定论，缺乏理论指导；

训练时，学习新样本有遗忘旧样本的趋势

常用改进方法

添加动量项，Dropout等规则化算法等；

采用改进的梯度下降法，使用硬件辅助计算；

RNN，LSTM等改进模型和神经元。

- 现状
- 神经网络
- **深度学习**
  - 介绍
  - 常见模型
    - Stacked Auto-Encoder
    - Convolutional Neural Network
    - Deep Belief Network
- 杂项

深度学习

- 深度学习的基础架构来自于前馈神经网络和BP算法，构造多层网络，通过最小化代价函数的方法来提高分类精度。
- 对于传统的ANN网络而言，由于多层网络训练的困难，实际使用的多数是只含有一层隐层节点的浅层模型。
- 深度学习更侧重于如何通过增加网络的深度，来减小每层需要拟合的特征个数，来提取出数据（尤其是语音与图像数据）的高层特征信息，从而达到更好的测试性能与分类精度。深度学习对输入数据逐级提取从底层到高层的特征，从而能更好地建立从底层信号到高层语义的复杂映射关系。
- 传统的机器学习方法，在训练数据量到达一定规模后，算法的学习能力就饱和了，而深度学习目前还看不到瓶颈。

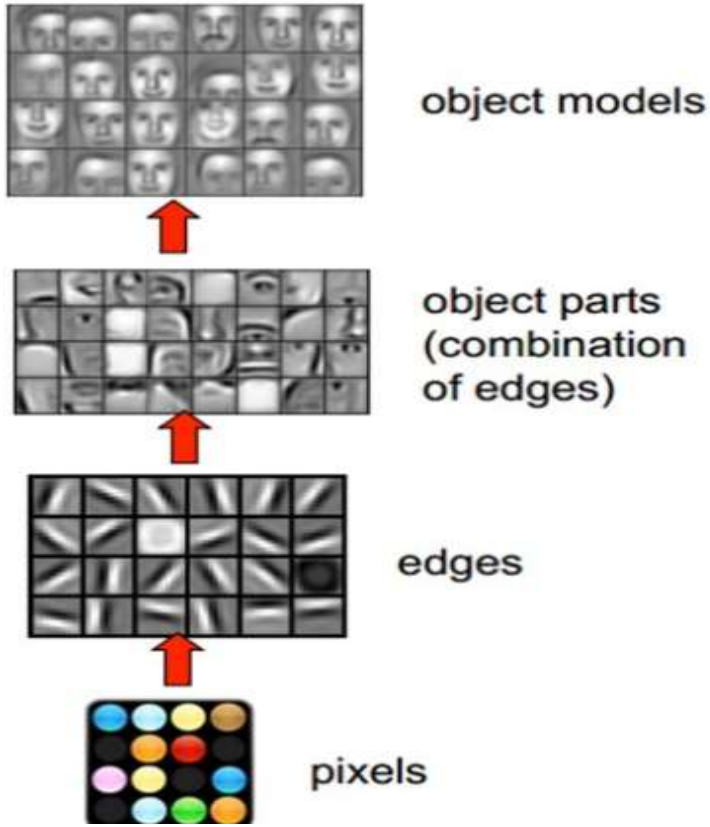
深度模型与浅层模型

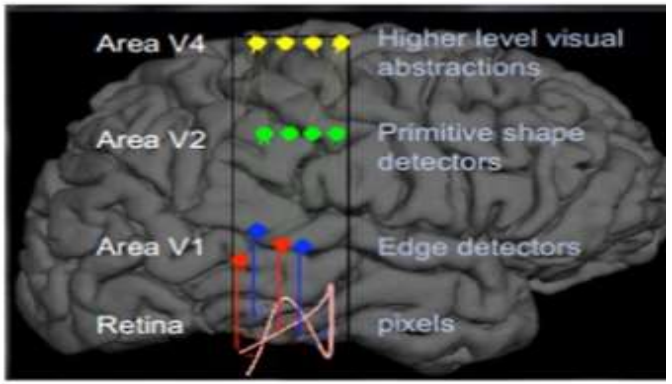
	浅层模型	深层模型
隐含层数	1-2	5层以上
表达能力	有限	更强
特征提取方式	人工设计或自动学习	自动学习
代价函数	存在凸代价函数，可以收敛到全局最优解	非凸代价函数，存在大量局部最优点。（实际情况是容易收敛到非常好的点）
训练难度	低	高
理论基础	成熟	不完善，需要根据实验结果调整模型
先验知识和工程经验	非常依赖	较少依赖
训练样本数量	多	非常多
使用场景	数据包含简单特征	数据包含复杂、多层、抽象特征

深度学习

- 深度学习的实质，是通过构建包含很多隐含层的神经网络模型，以更少的单层参数与更深的网络结构，通过海量训练数据，来学习特征的多层抽象表示，从而最终提升分类或预测的准确性。所以，**“深度模型”是手段**，**“特征学习”是目的**。区别于传统的浅层学习，深度学习的不同在于：
  - 1）强调了模型结构的深度，通常有5层以上、甚至100多层的隐含层；
  - 2）明确突出了特征学习的重要性，通过逐层特征变换，将样本在原空间的特征表示变换到一个新特征空间，使得分类或预测更加容易。

深层带来的好处





为什么采用层次网络

预训练与梯度消失现象

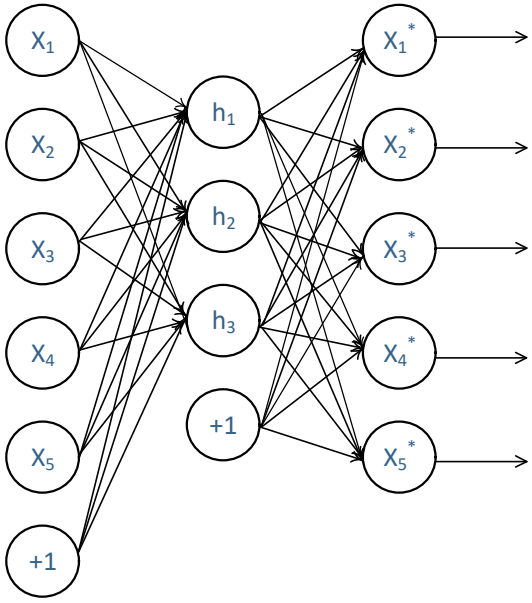
- 神经网络层数加深以后，容易出现梯度消失现象；
- 由于前层的梯度是由后层的梯度项相乘得到，梯度会逐层衰减，从而导致后层的网络学习率超过前层，BP算法收敛缓慢。当神经网络有很多层时，就会面临不稳定的情况。
- 对网络的预训练可以较好地避免这种现象。这是因为：

实验表明，在非凸优化问题上初始点的选择十分重要；
无监督学习增加了深层结构的鲁棒性；
对神经网络进行不同的预训练能够使网络学习到数据的不同的高质量特征；
单纯增加一个网络的深度，如果不进行预训练处理，会提高陷于局部极小点的可能性。

- 现状
- 神经网络
- **深度学习**
  - 介绍
  - **常见模型**
    - Stacked Auto-Encoder
    - Convolutional Neural Network
    - Deep Belief Network
- 杂项

自编码器结构

- 单层自动编码器网络（**AutoEncoder**）实质上是一个三层的反向传播神经网络。它逐层采用无监督学习的方式，不使用标签调整权值，将输入映射到隐含层上，再经过反变换映射到输出上，实现输入输出的近似等价。



- 自动编码器的主要思想是利用**无监督**方式最小化重建误差，学习到的权重提供了一个网络初始化的较好的初始点。无监督学习的主要目的是从无标签的数据中提取有用的特征，以减少输入信息，保留数据中关键的有效信息。网络通过没有标签的数据学习到潜在的分布信息，有利于它区分有标签的信息。然而，在网络中，权重仍然需要进行微调。因此，需要在神经网络的顶部增加一个线性回归，再对有标签的数据进行处理。网络的微调会采用梯度下降法，对所有层同时进行调整。

自编码器的建立

建立AutoEncoder的方法是：

对于m个数据的输入，有：

Code编码:使用非线性激活函数，将维输入数据映射到维隐含层（隐含节点表示特征）

$$h_i = f_\theta = \sigma(Wx_i + b), \theta = \{W, b\}$$

其中W是一个的权重矩阵，b是一个d'维的偏移向量

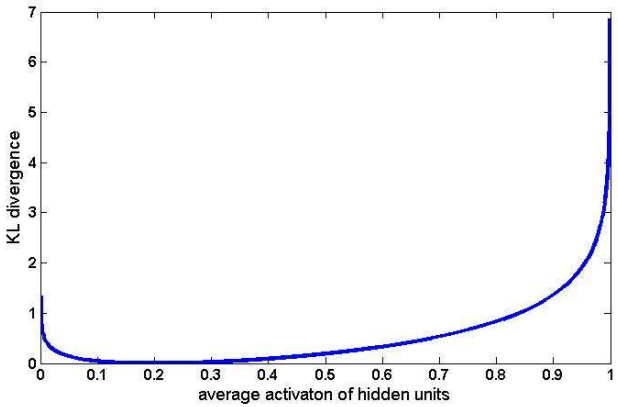
Decode解码:通过反向映射，对映射后的数据进行重建

$$y_i = f_{\theta'} = \sigma(W'h_i + b'), \theta' = \{W', b'\}$$

SAE网络每一次训练输入都会得到映射后的h_i 与解码后的y_i。通过对代价函数的最优化可以得到训练集上的权重与偏置。

节点的稀疏性限制

- 为了增强网络的稳定性，避免过拟合的发生，我们需要让少部分神经元输出值大于0，其他的大部分为0（或近似为0），这就是所谓的稀疏性。
- 在人脑中有大量的神经元，但是大多数自然图像通过视觉进入人脑时，只会刺激到少部分神经元，大部分神经元都是出于抑制状态的。而且，大多数自然图像，都可以被表示为少量基本元素（面或者线）的叠加。稀疏性处理能够更加有助于我们用少量的神经元提取出自然图像更加本质的特征。
- 从数学的角度来说，稀疏编码是一种多维数据描述方法，数据经稀疏编码后仅有少数分量同时处于明显激活状态。
- 在实际应用中，稀疏编码有如下几个优点：
  - 稀疏编码方案存储能力大，具有联想记忆能力，并且计算简便；
  - 使自然信号的结构更加清晰；
  - 事实上，这一简单的自编码神经网络通常可以学习出一个跟主成分分析（PCA）结果非常相似的输入数据的低维表示。

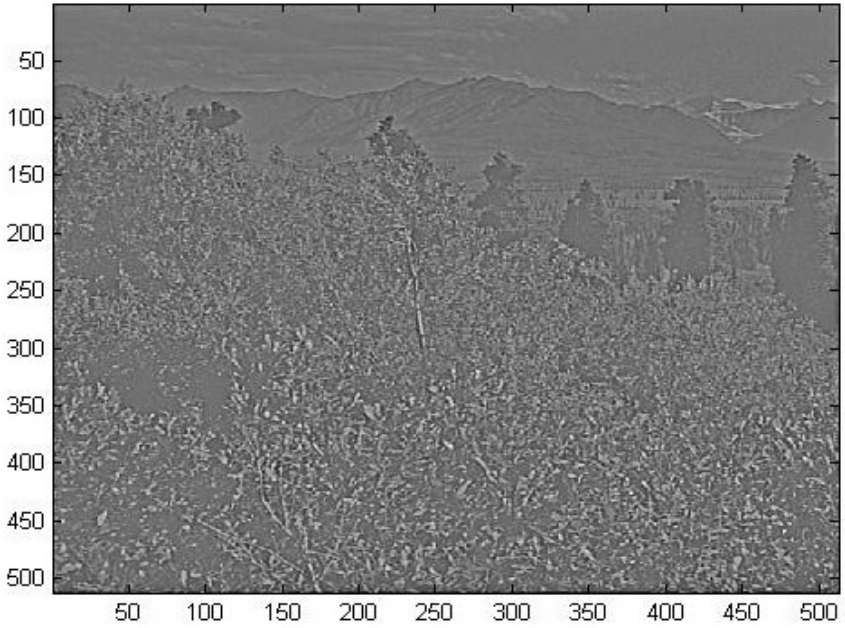


$$\sum_{j=1}^{s_2} KL(\rho \| \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

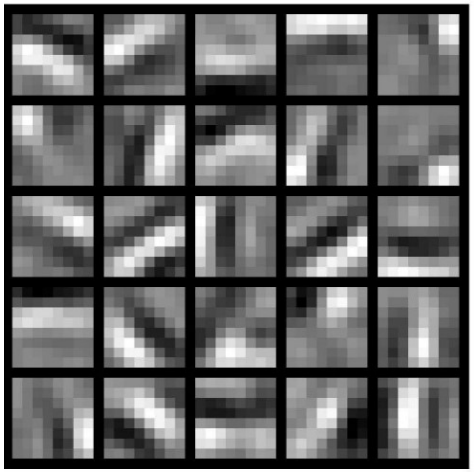
$$J_{sparse}(W,b) = J(\mathrm{W},\mathrm{b}) + \beta \sum_{j=1}^{s_2} KL(\rho \| \hat{\rho}_j)$$

图像实验

原始图像



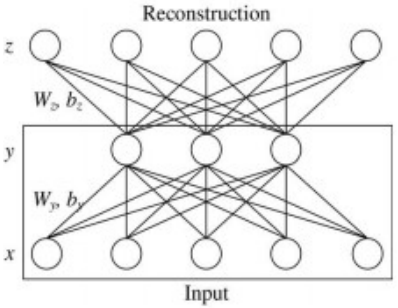
隐含层特征



可以看出，AE在图像处理的特征提取中可以作为边缘检测器学习到图像边缘信息

Stacked AutoEncoder

- 一个AE模型有1个可视层、1个隐含层1个重建层。
- 通过自下而上的映射，实现编码与反编码重建：



激活函数

$$y = f(x)$$

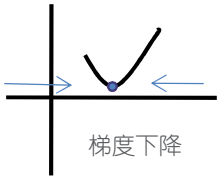
$$W = W - \eta \frac{\partial cost(x,z)}{\partial W}$$
$$b_y = b_y - \eta \frac{\partial cost(x,z)}{\partial b_y}$$
$$b_z = b_z - \eta \frac{\partial cost(x,z)}{\partial b_z}.$$

update

W, b_y, b_z

$$y = f(W_y x + b_y) \quad (1)$$
$$z = f(W_z y + b_z) \quad (2)$$

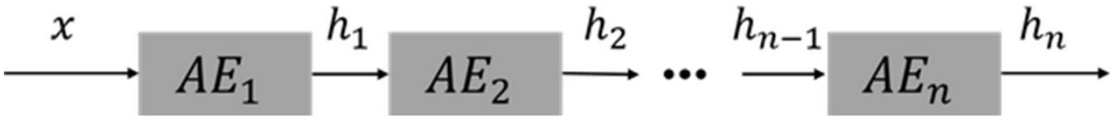
minimize



COST FUNCTION $\underset{W,b_y,b_z}{argmin}[c(x,z)]$

Stacked AutoEncoder 逐层训练

- **Stacked** 就是逐层堆叠的意思，这个跟“栈”有点像。当把多个自编码器 **Stack** 起来之后，这个系统看起来就像这样：

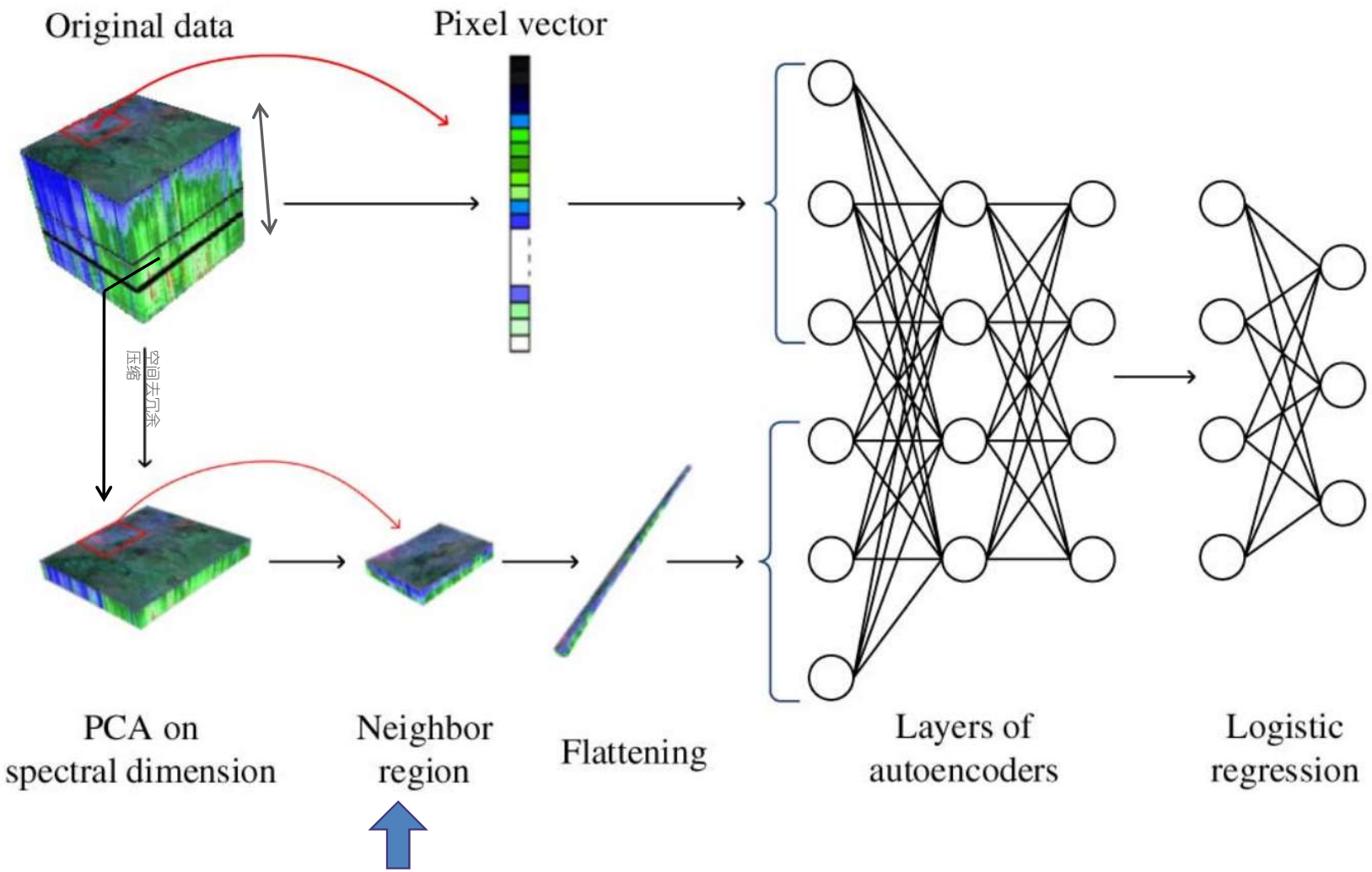


- 这样就把自编码器改成了深度结构了，即 **learning multiple levels of representation and abstraction (Hinton, Bengio, LeCun, 2015)**。需要注意的是，整个网络的训练不是一蹴而就的，而是逐层进行的。
- 比如说我们要训练一个 **n -> m -> k** 结构的网络，实际上我们是先训练网络 **n -> m -> n**，得到 **n -> m** 的变换，然后再训练 **m -> k -> m** 网络，得到 **m -> k** 的变换。最终堆叠成 **SAE**，即为 **n -> m -> k** 的结果，整个过程就像一层层往上面盖房子，这就是大名鼎鼎的 **layer-wise unsuperwised pre-training**（逐层非监督预训练）。

SAE构建方法

- 参数设置：设置好激活函数、学习率、迭代步数、训练层数等一系列基本参数；
- 构建SAE网络：分层建立输入-输出-输入的AE网络，并对权值与偏置初始化；
- SAE预训练：对网络只传入训练数据，在有限迭代步数下进行无监督的学习，以学得数据特征，得到权值与偏置的初始值；
- 参数微调：加入输出层，将SAE网络转化成输入-输出的NN网络并传入训练标签，即放弃输出到输入的反向映射，进行反向传播学习，减小训练误差；
- 测试：对测试数据进行神经网络测试，得到结果。

Example



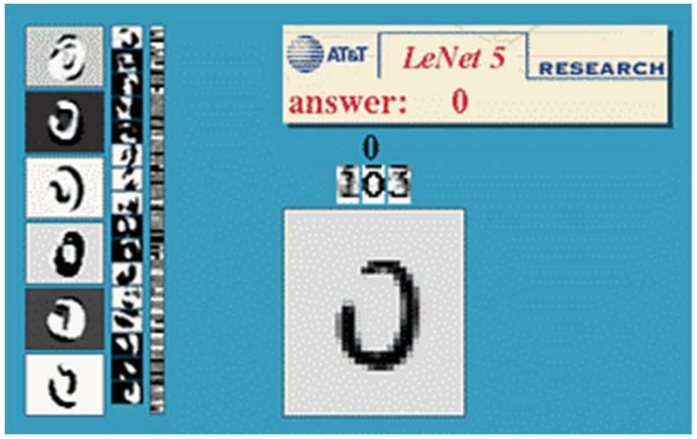
Defined By User

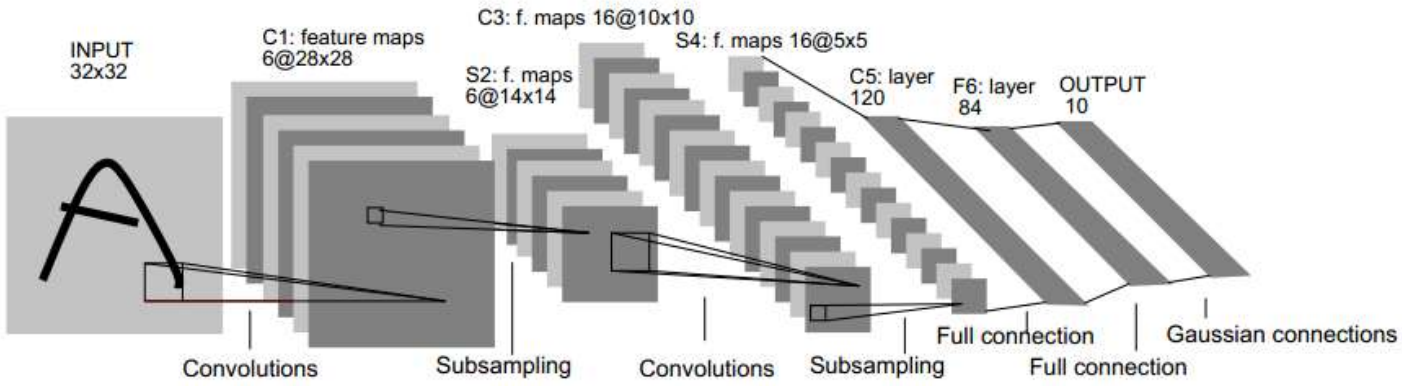
实验总结

- 1. 训练时间与迭代步数、层数、节点数、数据量四者呈线性关系；
- 2. 随着迭代步数的增加，分类精度迅速提高，然而分类有其上限，过多的迭代步数无法提高分类精度；不同的数据所适用的网络层数不一致，并不是层数越高，效果越理想；对节点的实验表明，神经网络更适用于数据的压缩，单层过高的节点数会降低分类效果；同时，数据量也比较明显地影响着分类精度；
- 3. 实验表明，向SAE网络中引入降维的数据并不能很好的提高分类效果；
- 4.对比其他分类算法SVM与KNN，实验表明，SAE网络更适合于大量数据的学习，而SVM与KNN浅层网络训练与测试所用时间更短，而且可以在较少的数据量下获得比较良好的分类效果。

CNN基本知识

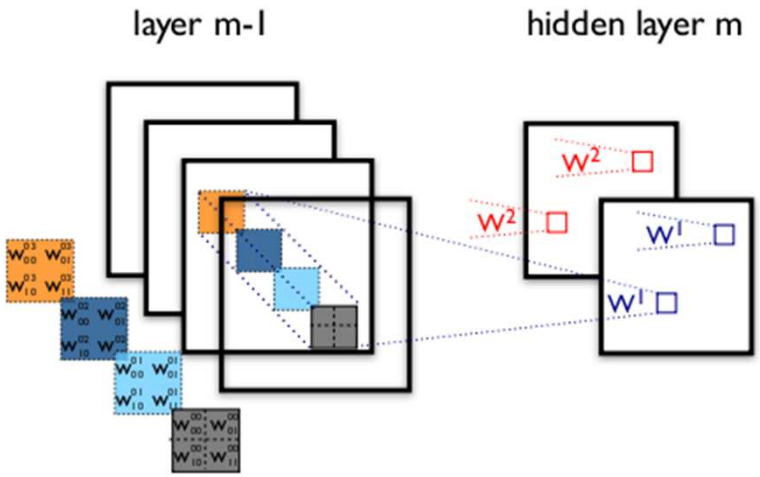
- 卷积神经网络是人工神经网络的一种，已成为当前图像识别领域的研究热点。卷积网络是为识别二维形状而特殊设计的一个多层感知器，这种网络结构对平移、比例缩放、旋转或者其他形式的变形具有一定的不变性。
- 卷积网络的核心思想是将：局部感知、权值共享以及下采样这三种结构结合起来，以达到图像降维的特征学习与分类。





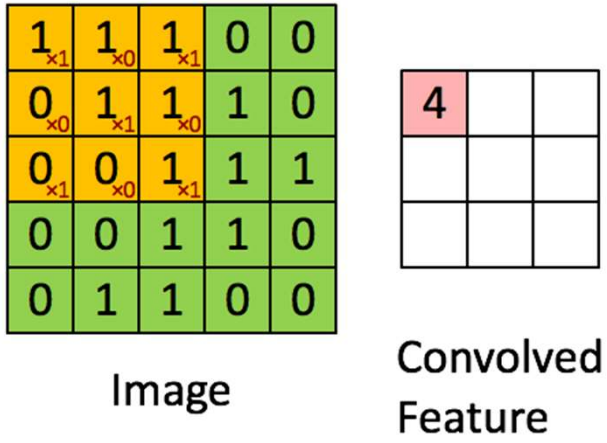
局部连接

- 卷积神经网络上下层之间采用局部连接的方式构建网络，每个隐含层的节点只与上一层的一部分连续输入点相连接，这样模拟了大脑皮层中，不同位置的视觉细胞只对局部区域有响应这一现象。
- **CNN**中局部连接以卷积的方法来实现。对于自然图像来说，从图像中某个部位提取到的特征也适用于其它部位，因此这种卷积和局部连接的特征提取方式可以提高特征的位置不变性。



权值共享

- 在卷积神经网络中，卷积层的每一个卷积核通过滑动窗口作用于整个输入图像，对输入图像进行卷积，卷积结果构成了输入图像的特征图，提取出图像的局部特征。每一个卷积滤波器共享相同的可学习参数，包括相同的卷积核权重矩阵和偏置项。
- 例如对于**32x32**的图像数据，全连接神经网络需要**32*32=1024**个神经元连接。采用卷积网权值共享方式，**32**个**3x3**的卷积核就可以了，一共**32x3x3=288**个可学习参数。

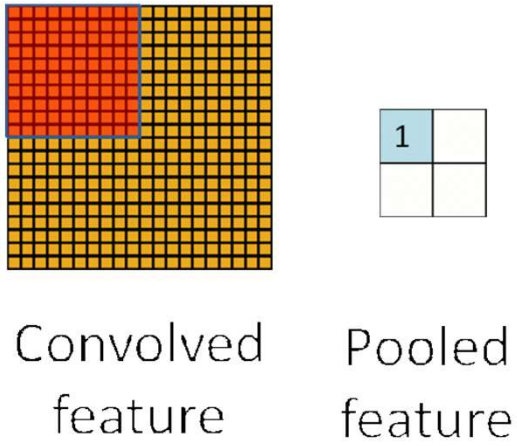


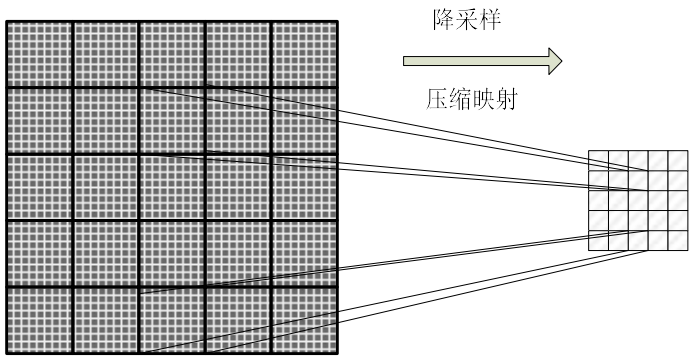
下采样

- 卷积核与输入图像进行卷积之后生成特征图，之后紧跟着一个进行二次筛选的下采样层（或称为池化层）。这种特有的两次特征提取结构使得网络对输入样本有较高的畸变容忍能力。
- 在通过卷积获得了特征 (features) 之后，下一步我们希望利用这些特征去做分类。理论上讲，人们可以用所有提取得到的特征去训练分类器，但这样做计算量太大。
- 例如：对于一个96X96像素的图像，假设我们已经学习得到了400个定义在8X8输入上的特征，每一个特征和图像卷积都会得到一个 (96 − 8 + 1) * (96 − 8 + 1) = 7921 维的卷积特征，由于有 400 个特征图，所以每个样例 (example) 都会得到一个 7921 * 400 = 3,168,400 维的特征向量。学习一个拥有超过3百万特征输入的分类器十分不便，并且容易出现过拟合 (over-fitting)。

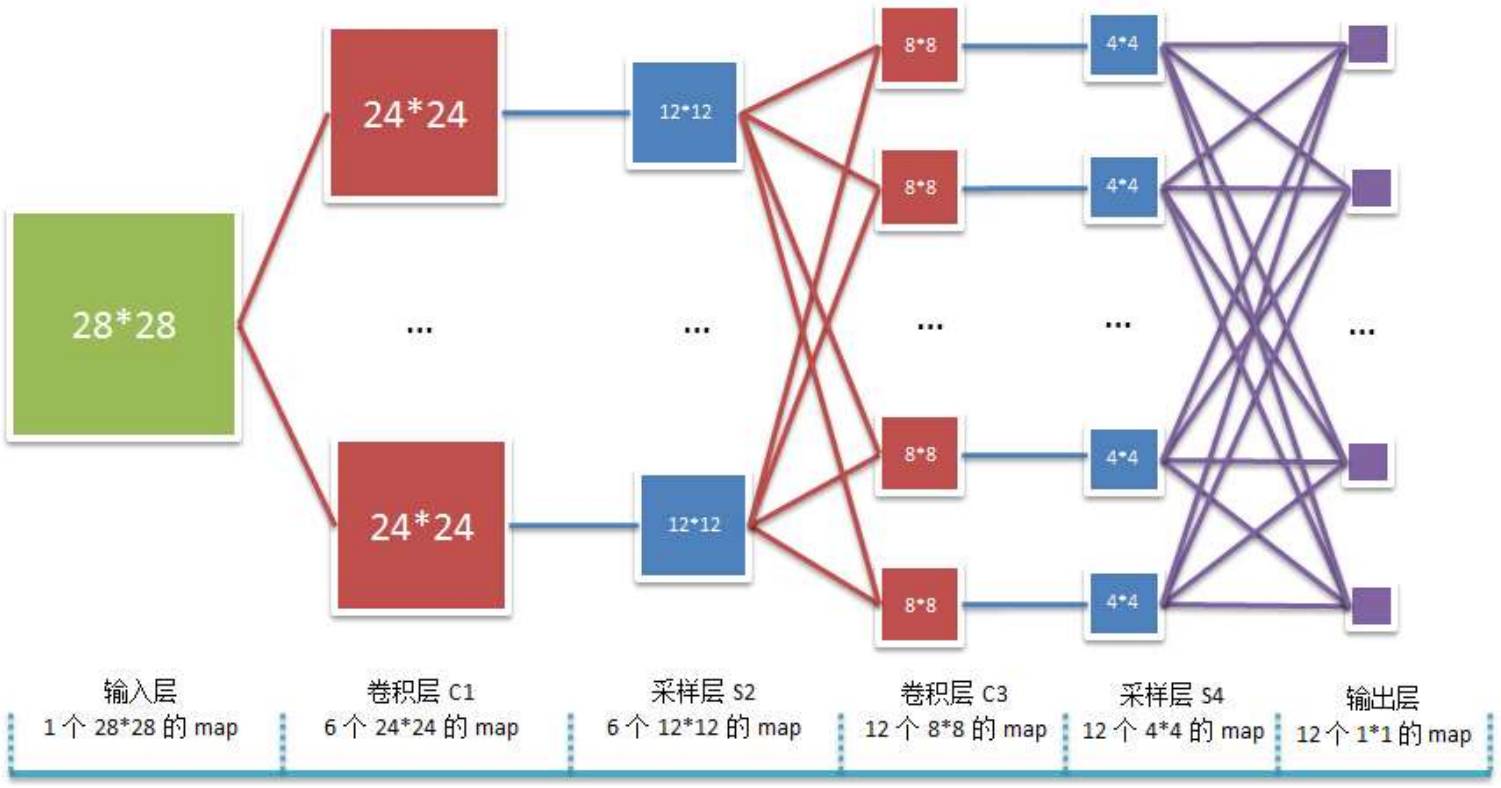
下采样

- 为了降低特征维度，一个很自然的想法就是对不同位置的特征进行聚合统计，例如，人们可以计算图像一个区域上的某个特定特征的平均值（或最大值）。这些概要统计特征不仅具有低得多的维度（相比使用所有提取得到的特征），同时还会改善结果(不容易过拟合)。这种聚合的操作就叫做池化（pooling），包括平均池化或者最大池等多种方法。





LeNet-5

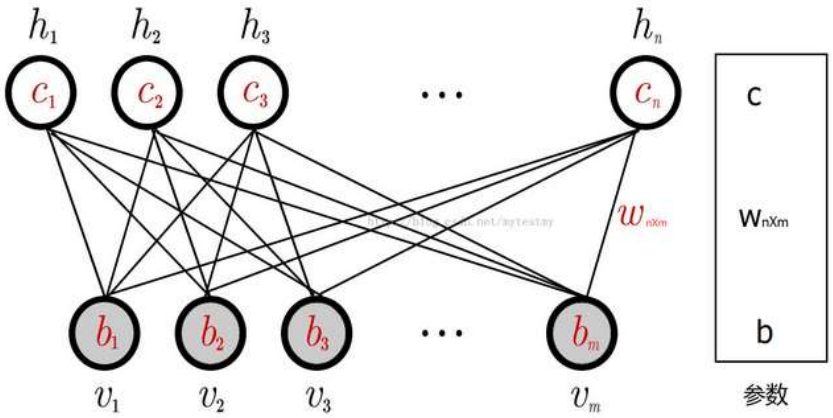

28*28
24*24
24*24
12*12
12*12
8*8
8*8
8*8
8*8
4*4
4*4
4*4
4*4
输入层
1 个 28*28 的 map
卷积层 C1
6 个 24*24 的 map
采样层 S2
6 个 12*12 的 map
卷积层 C3
12 个 8*8 的 map
采样层 S4
12 个 4*4 的 map
输出层
12 个 1*1 的 map

深度置信网络

- 深度置信网络（Deep Belief Net， DBN）是拥有深层架构的前馈神经网络，其中包含多个隐含层，而使用DBN的障碍在于如何训练这样的深层网络。
- 通常情况下，由于网络权值的随机初始化，基于梯度的优化容易陷入局部最小值。Hinton等提出了一种新的贪婪逐层非监督算法来初始化基于受限玻尔兹曼机（Restricted Boltzmann Machine， RBM）的DBN网络。
- 这个算法提供了网络权值的初始化方法，随后可以使用基于梯度的算法如梯度下降法来微调网络权值。

受限玻尔兹曼机

- RBM只有两层神经元，有m个可视节点和n个隐含节点，其中每个可视节点只和n个隐含节点相关，和其他可视节点是独立的。是即可视节点的状态只与n个隐含节点相关，对于每个隐含节点也是，只与m个可视节点相关，这个特点使得RBM的训练变得容易。注意这两层间的对称 双向连接。
- RBM网络有几个参数，一个是可视层与隐含层之间的权重矩阵，一个是可视节点的偏移量b，一个是隐含节点的偏移量c，这几个参数决定了RBM网络将一个m维的样本编码成一个什么样的n维的样本。



受限玻尔兹曼机

RBM介绍

- RBM区别于BM的一点是，前者要求层内神经元之间没有连接，而后者允许层内连接
- 定义能量函数 $$E_\theta(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} h_j w_{j,i} v_i,$$
- 联合概率分布 $$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{Z_\theta} e^{-E_\theta(\mathbf{v},\mathbf{h})} \qquad Z_\theta = \sum_{\mathbf{v},\mathbf{h}} e^{-E_\theta(\mathbf{v},\mathbf{h})}$$
- 性质：给定可见层时，隐含层神经元激活条件独立；反之亦然。

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n_h} P(h_j|\mathbf{v}); \quad P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{n_v} P(v_i|\mathbf{h}).$$

- 给定训练样本训练一个RBM意味着调整参数以拟合给定训练样本。数学推导可知，该问题等价于求下述函数最大值

$$\ln \mathcal{L}_{\theta,\mathbf{S}} = \ln \prod_{i=1}^{n_s} P(\mathbf{v}^i) = \sum_{i=1}^{n_s} \ln P(\mathbf{v}^i).$$

RBM训练

k 步 CD 算法 (简记为 CD-k) 的步骤很简单, 具体可描述为: 对 $\forall\ \mathbf{v} \in \mathbf{S}$, 取初始值 $\mathbf{v}^{(0)} := \mathbf{v}$, 然后执行 k 步 Gibbs 采样, 其中第 t 步 $(t = 1, 2, \cdots, k)$ 先后执行

- 利用 $P(\mathbf{h}|\mathbf{v}^{(t-1)})$ 采样出 $\mathbf{h}^{(t-1)}$;
- 利用 $P(\mathbf{v}|\mathbf{h}^{(t-1)})$ 采样出 $\mathbf{v}^{(t)}$,

学习率 η 较大时, 收敛速度更快, 但可能引起算法的不稳定, 而 η 较小时, 虽可避免不稳定情况的出现, 但收敛速度变慢. 为克服这一矛盾, 一种做法是在参数更新式中加入**动量项** (momentum), 使本次参数值修改的方向不完全由当前样本下似然函数的梯度方向决定, 而是采用上一次参数值修改方向与本次梯度方向的组合. 在某些情况下, 这可以避免算法过早地收敛到局部最优点. 具体格式可写为

$$\theta := \rho\theta + \eta\frac{\partial \ln \mathcal{L}_{\mathbf{S}}}{\partial \theta}. \tag{7.56}$$

- **输入**: 一个训练样本$\mathbf{x}_0$; 隐层单元个数m; 学习率ϵ; 最大训练周期T.
- **输出**: 连接权重矩阵W、可见层的偏置向量$\mathbf{a}$、隐层的偏置向量$\mathbf{b}$.
- **训练阶段**:
  初始化: 令可见层单元的初始状态$\mathbf{v}_1 = \mathbf{x}_0$; W、$\mathbf{a}$和$\mathbf{b}$为随机的较小数值。
  For $t = 1, 2, \cdots, T$
    For $j = 1, 2, \cdots, m$(对所有隐单元)
      计算$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1)$, 即$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1) = \sigma(b_j + \sum_i v_{1i}W_{ij})$;
      从条件分布$P(\mathbf{h}_{1j}|\mathbf{v}_1)$中抽取$\mathbf{h}_{1j} \in \{0, 1\}$.
    EndFor
    For $i = 1, 2, \cdots, n$(对所有可见单元)
      计算$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1)$, 即$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1) = \sigma(a_i + \sum_j W_{ij}h_{1j})$;
      从条件分布$P(\mathbf{v}_{2i}|\mathbf{h}_1)$中抽取$\mathbf{v}_{2i} \in \{0, 1\}$.
    EndFor
    For $j = 1, 2, \cdots, m$(对所有隐单元)
      计算$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2)$, 即$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2) = \sigma(b_j + \sum_i v_{2i}W_{ij})$;
    EndFor
  按下式更新各个参数
  - $W \leftarrow W + \epsilon(P(\mathbf{h}_{1\cdot} = 1|\mathbf{v}_1)\mathbf{v}_1^T - P(\mathbf{h}_{2\cdot} = 1|\mathbf{v}_2)\mathbf{v}_2^T)$;
  - $\mathbf{a} \leftarrow \mathbf{a} + \epsilon(\mathbf{v}_1 - \mathbf{v}_2)$;
  - $\mathbf{b} \leftarrow \mathbf{b} + \epsilon(P(\mathbf{h}_{1\cdot} = 1|\mathbf{v}_1) - P(\mathbf{h}_{2\cdot} = 1)|\mathbf{v}_2)$;

- 一般地，链接权重W_{ij}可初始化为来自正态分布N(0,0.01)的随机数，隐单元的偏置c_j初始化为0；
- 对于第i个可见单元，偏置b_i初始化为log[pi/(1-pi)] 。pi表示训练样本中第i个特征处于激活状态所占的比率
- 学习率epsilon至关重要，大则收敛快，但是算法可能不稳定。小则慢。为克服这一矛盾引入动量，使本次参数值修改的方向不完全由当前样本似然函数梯度方向决定，而是上一次参数值修改方向与本次梯度方向的结合可以避免过早的收敛到局部最优点
- k为动量项学习率 开始时k=0.5，重构误差处于平稳增加状态时，k=0.9

按下式更新各个参数

- $W \leftarrow W + \epsilon(P(\mathbf{h}_{1\cdot} = 1|\mathbf{v}_1)\mathbf{v}_1^T - P(\mathbf{h}_{2\cdot} = 1|\mathbf{v}_2)\mathbf{v}_2^T)$;
- $\mathbf{a} \leftarrow \mathbf{a} + \epsilon(\mathbf{v}_1 - \mathbf{v}_2)$;
- $\mathbf{b} \leftarrow \mathbf{b} + \epsilon(P(\mathbf{h}_{1\cdot} = 1|\mathbf{v}_1) - P(\mathbf{h}_{2\cdot} = 1)|\mathbf{v}_2)$;

记号$P(\mathbf{h}_{k\cdot} = 1|\mathbf{v}_k)$ $(k = 1, 2)$是m维列向量, 其第j个元素为$P(\mathbf{h}_{kj} = 1|\mathbf{v}_k)$

$$W_{ij}^{(t+1)} = kW_{ij}^{(t)} + \epsilon\frac{\partial\mathcal{L}}{\partial W_{ij}^{(t)}},$$

RBM评估

常用的近似方法是**重构误差**, 所谓重构误差 (reconstruction error), 就是以训练样本作为初始状态, 经过 RBM 的分布进行一次 Gibbs 转移后与原数据的差异量, 具体可这样计算

error $= 0$
FORALL $\mathbf{v} \in \mathbf{S}$ **DO**
{
1. $\mathbf{h} =$ sample_h_given_v($\mathbf{v}$, RBM(W, $\mathbf{a}$, $\mathbf{b}$))
2. $\tilde{\mathbf{v}} =$ sample_v_given_h($\mathbf{h}$, RBM(W, $\mathbf{a}$, $\mathbf{b}$))
3. $error = error + \|\mathbf{v} - \tilde{\mathbf{v}}\|$.

}

其中关于函数 sample_h_given_v 和 sample_v_given_h 的定义参见 §6, 范数 $\|\cdot\|$ 采用 1- 范数或 2- 范数 (以 $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ 为例, $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$, $\|\mathbf{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$).

重构误差能够在一定程度上反映 RBM 对训练样本的似然度, 不过并不完全可靠. 但总的来说, 其计算相当简单, 因此在实践中非常有用.

目录
CONTENTS

- 现状
- 神经网络
- 深度学习
  - 介绍
  - 常见模型
    - Stacked Auto-Encoder
    - Convolutional Neural Network
    - Deep Belief Network
- 杂项

对深度学习的评价

- 理论上无法证明它为何有效，但是在实际问题上它的确有效。没有形成系统的理论。**各个DL模型为什么好用？原理本质是什么？各个模型都适用于什么场合？针对特定数据，特定问题，如何组合搭建模型，各个参数怎么选？如何根据特定模型，特定数据来训练模型？**
- 优点：1.在计算机视觉和语音识别方面确实结果超过传统方法一大截；2.具有较好的transfer learning性质，一个模型训练好了拿到另一个问题上做一些简单的refinement就可以用了；3.**只要层数够**，就可以学到足够好的特征。
- 缺点：1.训练耗时，模型正确性验证复杂且麻烦；2. 某些深度网络需要GPU支持；3.某些**模型难以诠释**，学得的特征**对人而言并不直观**。

应用

- 图像识别，文本模型，时序相关数据序列
- 语音识别+机器翻译
- **Multimodal Learning**
- 广告系统
- 把深度学习思想应用到其他的模型上

展望

- 当前深度学习领域的学术研究可以包含四部分：优化，泛化，表达以及应用。除了应用之外每个部分又可以分成实践和理论两个方面；
- 包括谷歌、百度等都在研究，并产生了很多理论研究与工程应用；
- 深度学习引领了机器学习的方向，相关的机器视觉方向，从理论到产品，导致了这个行业不断的火热。

参考资料

- **Stanford Tutorial**

  http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial

  http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning

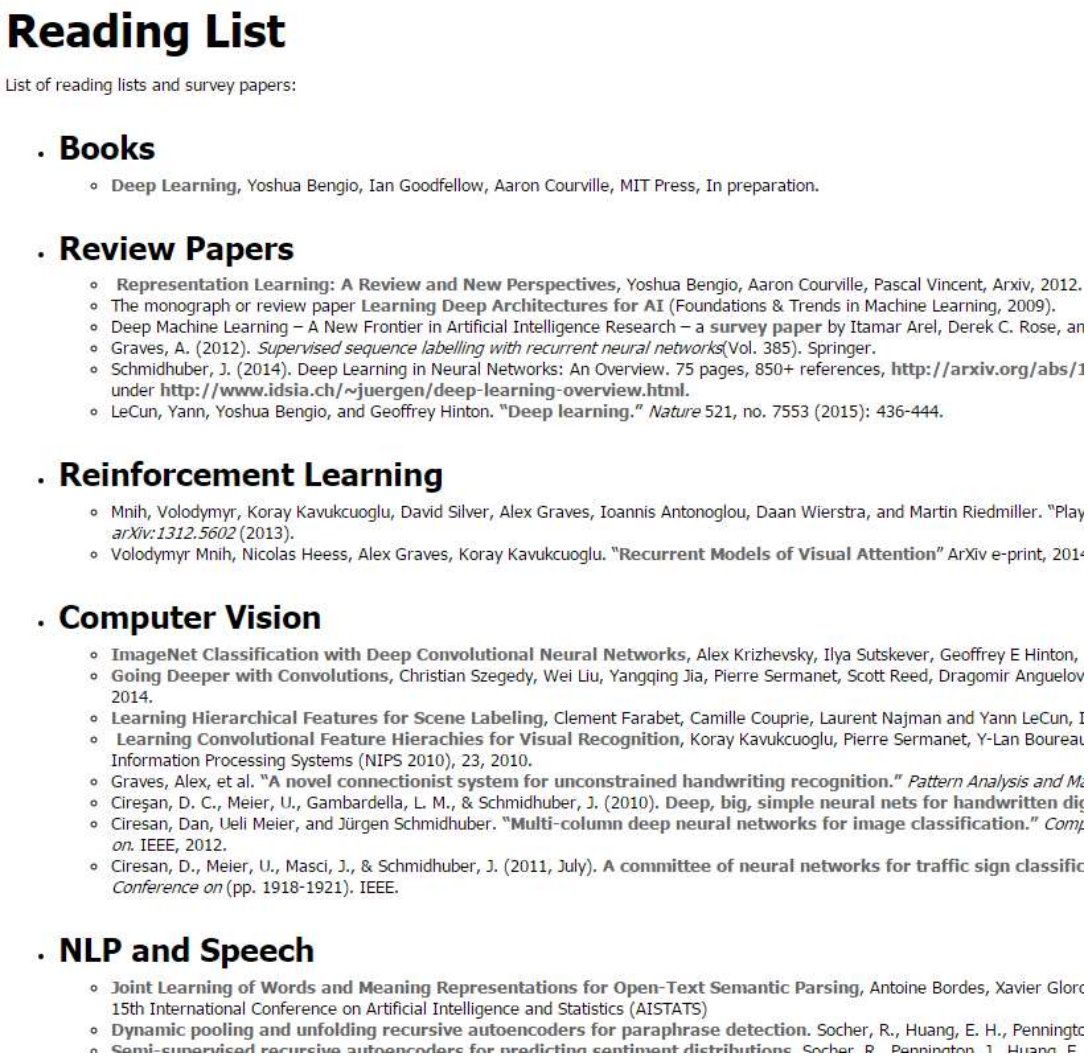
- **http://www.deeplearning.net/**

  - a **reading list**,
  - links to **software**,
  - **datasets**,
  - a list of **deep learning research groups and labs**,
  - a list of announcements for deep learning related jobs (**job listings**),
  - as well as **tutorials** and cool **demos**.
  - announcements and news about **deep learning**

参考资料

- http://www.deeplearning.net/

- 以及Transfer Learning, Practical Tricks and Guides, Sparse Coding, Foundation Theory and Motivation, Unsupervised Feature Learning, Autoencoders等多个专题。



Reading List

List of reading lists and survey papers:

Books

- **Deep Learning**, Yoshua Bengio, Ian Goodfellow, Aaron Courville, MIT Press, In preparation.

Review Papers

- **Representation Learning: A Review and New Perspectives**, Yoshua Bengio, Aaron Courville, Pascal Vincent, Arxiv, 2012.
- The monograph or review paper **Learning Deep Architectures for AI** (Foundations & Trends in Machine Learning, 2009).
- Deep Machine Learning – A New Frontier in Artificial Intelligence Research – a **survey paper** by Itamar Arel, Derek C. Rose, an
- Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*(Vol. 385). Springer.
- Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. 75 pages, 850+ references, **http://arxiv.org/abs/1** under **http://www.idsia.ch/~juergen/deep-learning-overview.html**.
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "**Deep learning.**" *Nature* 521, no. 7553 (2015): 436-444.

Reinforcement Learning

- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Play *arXiv:1312.5602* (2013).
- Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu. "**Recurrent Models of Visual Attention**" ArXiv e-print, 201

Computer Vision

- **ImageNet Classification with Deep Convolutional Neural Networks**, Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton,
- **Going Deeper with Convolutions**, Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov 2014.
- **Learning Hierarchical Features for Scene Labeling**, Clement Farabet, Camille Couprie, Laurent Najman and Yann LeCun, I
- **Learning Convolutional Feature Hierachies for Visual Recognition**, Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau Information Processing Systems (NIPS 2010), 23, 2010.
- Graves, Alex, et al. "**A novel connectionist system for unconstrained handwriting recognition.**" *Pattern Analysis and M*
- Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). **Deep, big, simple neural nets for handwritten dig**
- Ciresan, Dan, Ueli Meier, and Jürgen Schmidhuber. "**Multi-column deep neural networks for image classification.**" *Comp on.* IEEE, 2012.
- Ciresan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011, July). **A committee of neural networks for traffic sign classific** *Conference on* (pp. 1918-1921). IEEE.

NLP and Speech

- **Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing**, Antoine Bordes, Xavier Glorc 15th International Conference on Artificial Intelligence and Statistics (AISTATS)
- **Dynamic pooling and unfolding recursive autoencoders for paraphrase detection.** Socher, R., Huang, E. H., Penningtc
- **Semi-supervised recursive autoencoders for predicting sentiment distributions.** Socher, R., Pennington, J., Huang, E.

参考资料

- http://www.deeplearning.net/， 43个软件

Software links

1. **Theano** – CPU/GPU symbolic expression compiler in python (from MILA lab at University of Montreal)
2. **Torch** – provides a Matlab-like environment for state-of-the-art machine learning algorithms in lua (from Ron
3. **Pylearn2** - Pylearn2 is a library designed to make machine learning research easy.
4. **Blocks** - A Theano framework for training neural networks
5. **Tensorflow** - TensorFlow™ is an open source software library for numerical computation using data flow gra
6. **MXNet** - MXNet is a deep learning framework designed for both efficiency and flexibility.
7. **Caffe** -Caffe is a deep learning framework made with expression, speed, and modularity in mind.Caffe is a d
8. **Lasagne** - Lasagne is a lightweight library to build and train neural networks in Theano.
9. **Keras**- A theano based deep learning library.
10. **Deep Learning Tutorials** – examples of how to *do* Deep Learning with Theano (from LISA lab at University
11. **Chainer** - A GPU based Neural Network Framework
12. **CNTK** – Computational Network Toolkit – is a unified deep-learning toolkit by Microsoft Research.
13. **MatConvNet** - A MATLAB toolbox implementing Convolutional Neural Networks (CNNs) for computer vision a
14. **DeepLearnToolbox** – A Matlab toolbox for Deep Learning (from Rasmus Berg Palm)
15. **Cuda-Convnet** – A fast C++/CUDA implementation of convolutional (or more generally, feed-forward) neura
    graph of layers will do. Training is done using the back-propagation algorithm.
16. **Deep Belief Networks**. Matlab code for learning Deep Belief Networks (from Ruslan Salakhutdinov).
17. **RNNLM**- Tomas Mikolov's Recurrent Neural Network based Language models Toolkit.
18. **RNNLIB**-RNNLIB is a recurrent neural network library for sequence learning problems. Applicable to most typ
    recognition.

参考资料

- http://www.deeplearning.net/

- 大量数据集

Datasets

These datasets can be used for benchmarking deep learning algorithms:

Symbolic Music Datasets

- Piano-midi.de: classical piano pieces (**http://www.piano-midi.de/**)
- Nottingham : over 1000 folk tunes (**http://abc.sourceforge.net/NMD/**)
- MuseData: electronic library of classical music scores (**http://musedata.stanford.edu/**)
- JSB Chorales: set of four-part harmonized chorales (**http://www.jsbchorales.net/index.shtml**)

Natural Images

- MNIST: handwritten digits (**http://yann.lecun.com/exdb/mnist/**)
- NIST: similar to MNIST, but larger
- Perturbed NIST: a dataset developed in Yoshua's class (NIST with tons of deformations)
- CIFAR10 / CIFAR100: 32×32 natural image dataset with 10/100 categories (**http://www.cs.utoronto**
- Caltech 101: pictures of objects belonging to 101 categories (**http://www.vision.caltech.edu/Image**
- Caltech 256: pictures of objects belonging to 256 categories (**http://www.vision.caltech.edu/Image**
- Caltech Silhouettes: 28×28 binary images contains silhouettes of the Caltech 101 dataset
- STL-10 dataset is an image recognition dataset for developing unsupervised feature learning, deep learn modifications. **http://www.stanford.edu/~acoates//stl10/**
- The Street View House Numbers (SVHN) Dataset - **http://ufldl.stanford.edu/housenumbers/**
- NORB: binocular images of toy figurines under various illumination and pose (**http://www.cs.nyu.edu**
- Imagenet: image database organized according to the WordNethierarchy (**http://www.image-net.or**
- Pascal VOC: various object recognition challenges (**http://pascallin.ecs.soton.ac.uk/challenges/V**
- Labelme: A large dataset of annotated images, **http://labelme.csail.mit.edu/Release3.0/browser**
- COIL 20: different objects imaged at every angle in a 360 rotation(**http://www.cs.columbia.edu/CA**
- COIL100: different objects imaged at every angle in a 360 rotation (**http://www1.cs.columbia.edu/C**

Artificial Datasets

- **Arcade Universe** - An artificial dataset generator with images containing arcade games sprites such as generator.
- A collection of datasets inspired by the ideas from **BabyAISchool**:
  - **BabyAIShapesDatasets** : distinguishing between 3 simple shapes
  - **BabyAIImageAndQuestionDatasets** : a question-image-answer dataset
- Datasets generated for the purpose of an empirical evaluation of deep architectures (**DeepVsShallowC**
  - **MnistVariations** : introducing controlled variations in MNIST
  - **RectanglesData** : discriminating between wide and tall rectangles
  - **ConvexNonConvex** : discriminating between convex and nonconvex shapes
  - **BackgroundCorrelation** : controlling the degree of correlation in noisy MNIST backgrounds

Faces

- Labelled Faces in the Wild: 13,000 images of faces collected from the web, labelled with the name of the
- Toronto Face Dataset

Thanks