11-深度学习

武汉大学计算机学院本科生课程

模式识别与机器学习

(PR & ML)

武汉大学计算机学院

Email: 18986211797@189.cn

Ch 7　深度学习

7.1 深度学习简介

7.2 深度学习训练过程

7.3 深度学习常用模型及方法

(重点学习掌握DL思想方法：自动编码器AutoEncoder，卷积神经网络CNN，经典的CNN举例：LeNet-5网络结构)

7.4 深度学习的应用

7.5 深度学习展望

7.1 深度学习简介

深度学习(Deep Learning)是一种基于无监督特征学习和特征层次结构的学习方法。可能的名称还有：特征学习或无监督特征学习。

------学习深度学习要具备一定的神经网络知识：一般需先学习掌握传统的人工神经网络（主要有感知器、BP神经网络等）的基础知识，再学习研究“深度学习”相关部分。

图　深度学习、机器学习、人工智能三者关系

在机器学习中，获得好的特征是识别成功的关键。

1. 为什么要自动学习特征

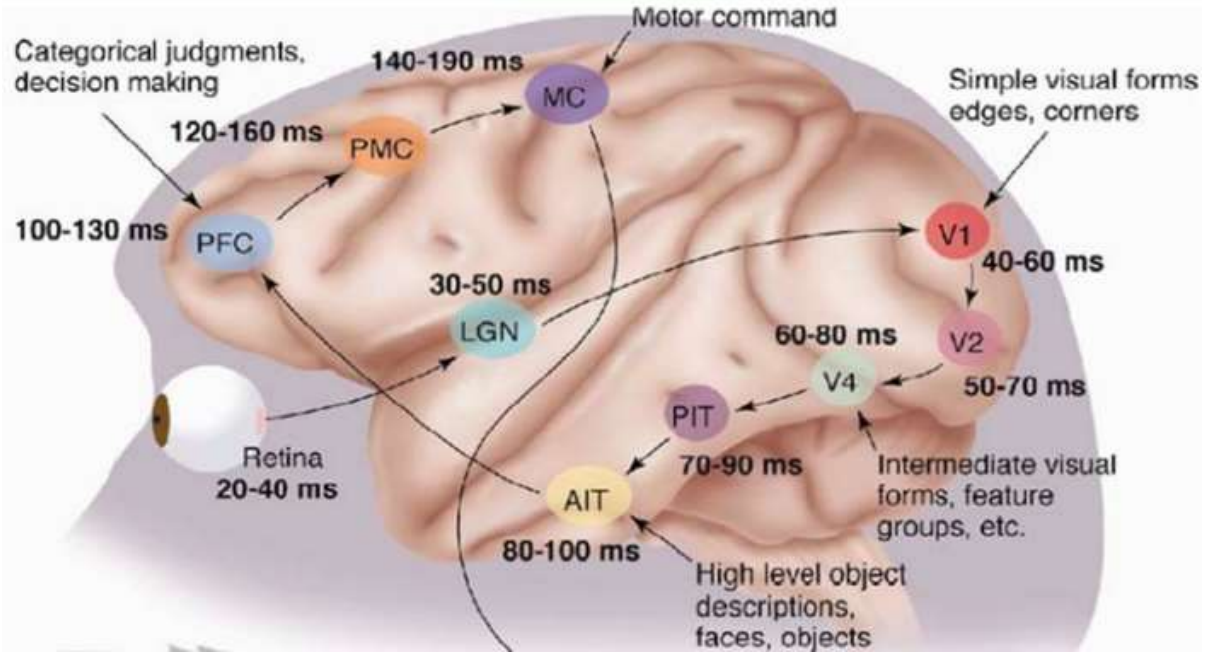
· 一般而言，特征越多，给出信息就越多，识别准确性会得到提升；
· 但特征多，计算复杂度增加，探索的空间大，可以用来训练的数据在每个特征上就会稀疏。
· 结论：不一定特征越多越好！需要有多少个特征，需要学习确定。

2. 为什么采用层次网络结构

■人脑视觉机理

√1981年的诺贝尔医学奖获得者 David Hubel和Torsten Wiesel发现了视觉系统的信息处理机制；

√发现了一种被称为“方向选择性细胞的神经元细胞，当瞳孔发现了眼前的物体的边缘，而且这个边缘指向某个方向时，这种神经元细胞就会活跃。



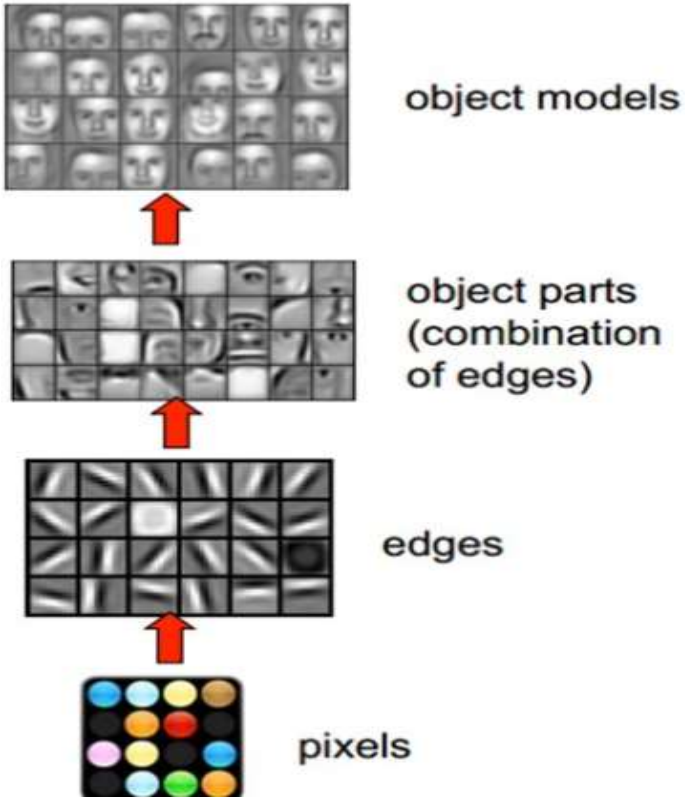
2. 为什么采用层次网络结构

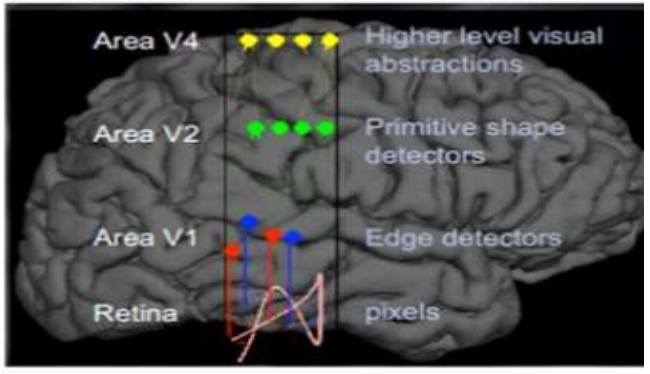
■人脑视觉机理

√人的视觉系统的信息处理是分级的；

√高层的特征是低层特征的组合，从低层到高层的特征表示越来越抽象，越来越能表现语义或者意图；

√抽象层面越高，存在的可能猜测就越少，就越有利于分类。





2. 为什么采用层次网络结构

■浅层学习的局限

√人工神经网络（BP算法）

—虽被称作多层感知器，但实际应用中基本上是只含有一层隐层节点的浅层模型

√SVM、Boosting、最大熵方法（如LR：Logistic Regression）

—带有一层隐层节点（如SVM、Boosting），或没有隐层节点（如LR）的浅层模型

局限性：有限样本和计算单元情况下对复杂函数的表示能力有限，针对复杂分类问题其泛化能力受限。

深度学习

- 2006年，加拿大多伦多大学教授、机器学习领域的泰斗Geoffrey Hinton在《Science》上发表论文提出深度学习主要观点[1]：

1）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，从而有利于可视化或分类；

2）深度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。

Reference

[1] G. E. Hinton, et al. Reducing the Dimensionality of Data with Neural Networks. Science 28 July, Pages: 504-507, 2006.

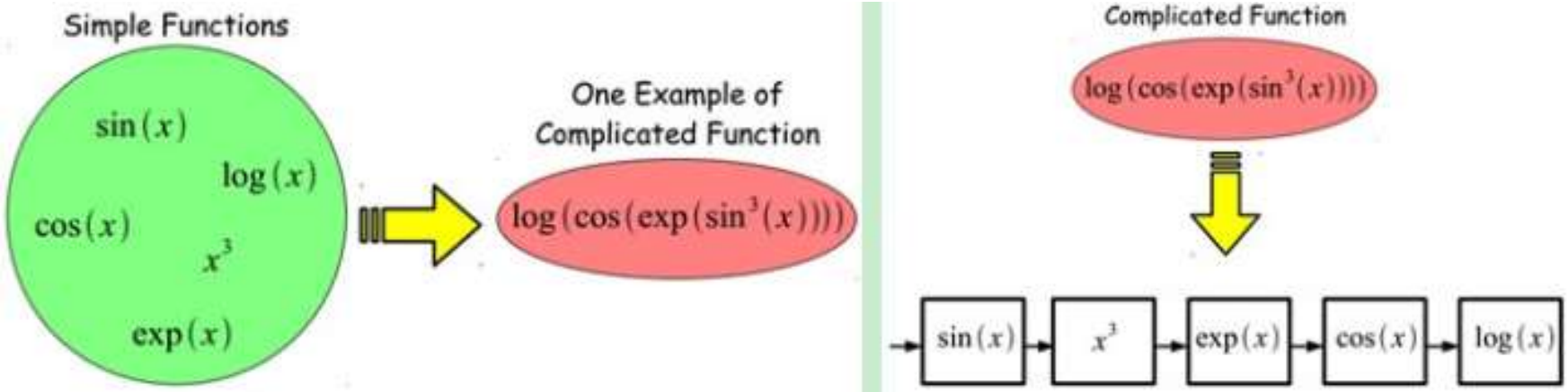
深度学习

- **本质**：深度学习的实质，是通过构建具有很多隐层的机器学习模型和海量的训练数据，来学习更有用的特征，从而最终提升分类或预测的准确性。**“深度模型”是手段，“特征学习”是目的。**
- **与浅层学习(shallow learning)区别**：

1) 强调了模型结构的深度，通常有5-10多层的隐层节点；
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，从而使分类或预测更加容易。**与人工规则构造特征的方法相比，利用大数据来学习特征，更能够刻画数据的丰富内在信息。**

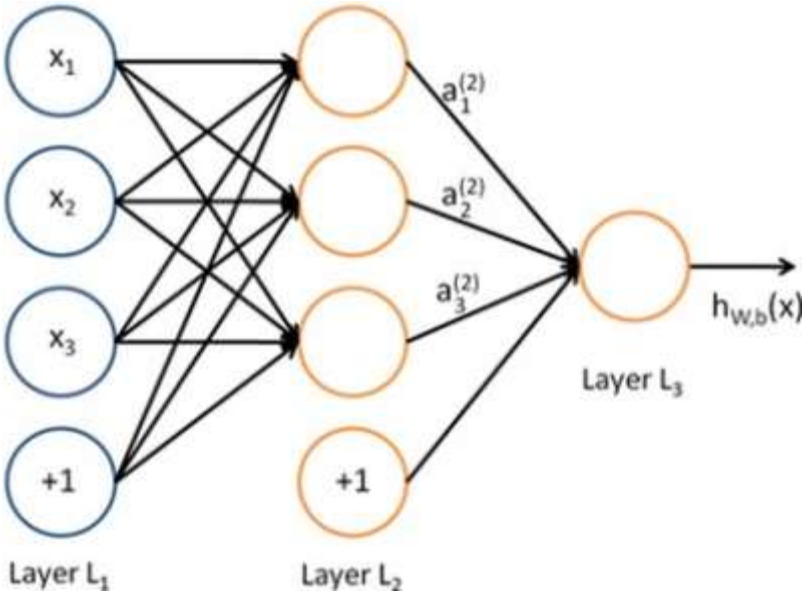
深度学习

- 优点：可通过学习一种深层非线性网络结构，实现复杂函数逼近，表征输入数据分布式表示。

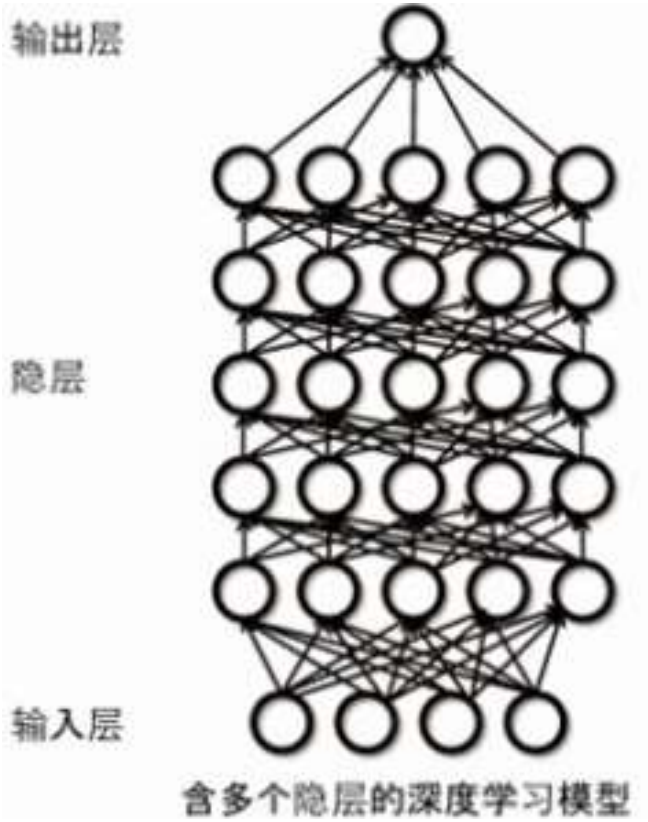


深度学习 vs. 神经网络

神经网络：



深度学习：



深度学习 vs. 神经网络

相同点：二者均采用分层结构，系统包括输入层、隐层（多层）、输出层组成的多层网络，只有相邻层节点之间有连接，同一层以及跨层节点之间相互无连接，每一层可以看作是一个Logistic 回归模型。

不同点：

神经网络：采用BP算法调整参数，即采用迭代算法来训练整个网络。随机设定初值，计算当前网络的输出，然后根据当前输出和样本真实标签之间的差去改变前面各层的参数，直到收敛；

深度学习：采用逐层训练机制。采用该机制的原因在于如果采用BP机制，对于一个deep network（7层以上），残差传播到最前面的层将变得很小，出现所谓的gradient diffusion（梯度扩散）。

深度学习 vs. 神经网络

- 神经网络的局限性：

1)较容易过拟合，参数较难调整，而且需要不少技巧；

2)训练速度较慢，在层次比较少(小于等于3)的情况下效果并不比其它方法更优；

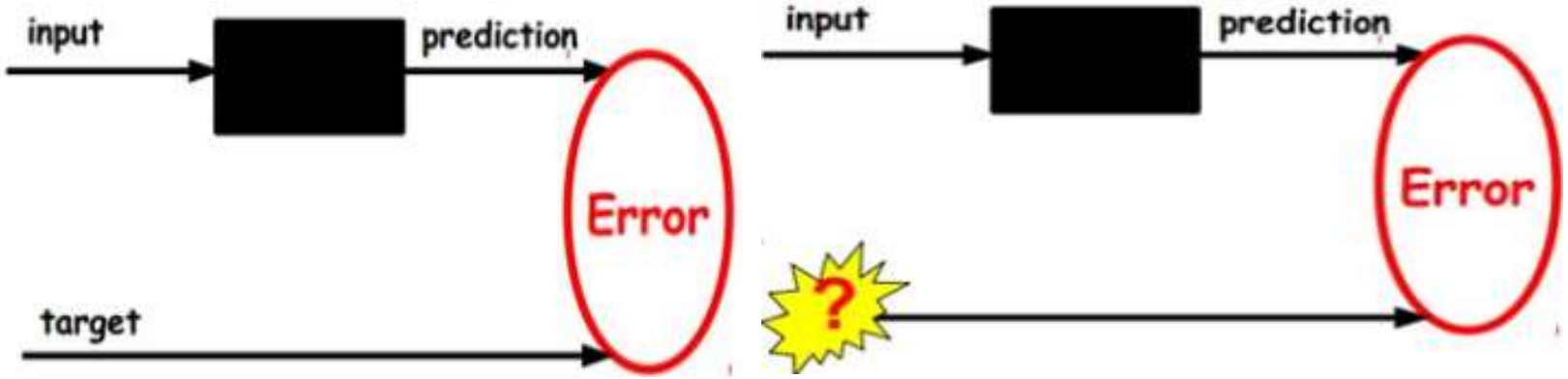
7.2 深度学习训练过程

- 不采用BP算法的原因

（1）反馈调整时，梯度越来越稀疏，从顶层越往下，误差校正信号越来越小；

（2）收敛易陷入局部极小，由于是采用随机值初始化，当初值是远离最优区域时易导致这一情况；

（3）BP算法需要有标签数据来训练，但大部分数据是无标签的；



7.2 深度学习训练过程

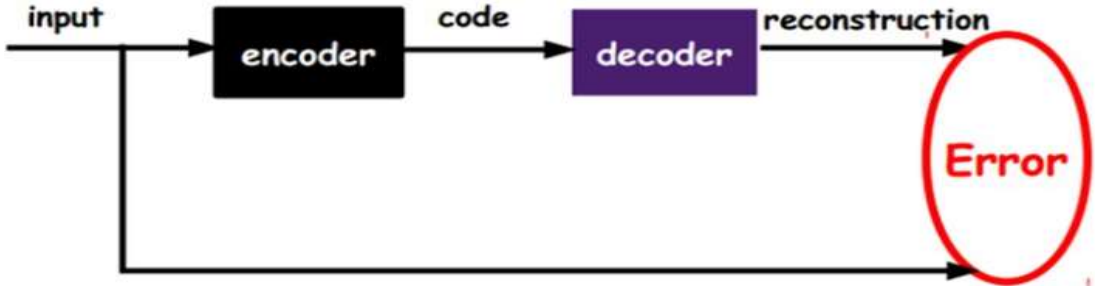
- 第一步：采用自下而上的无监督学习

1）逐层构建单层神经元。

2）每层采用wake-sleep算法进行调优。每次仅调整一层，逐层调整。

该过程可以看作是一个feature learning的过程，是和传统神经网络区别最大的部分。

7.2 深度学习训练过程



- wake-sleep算法：

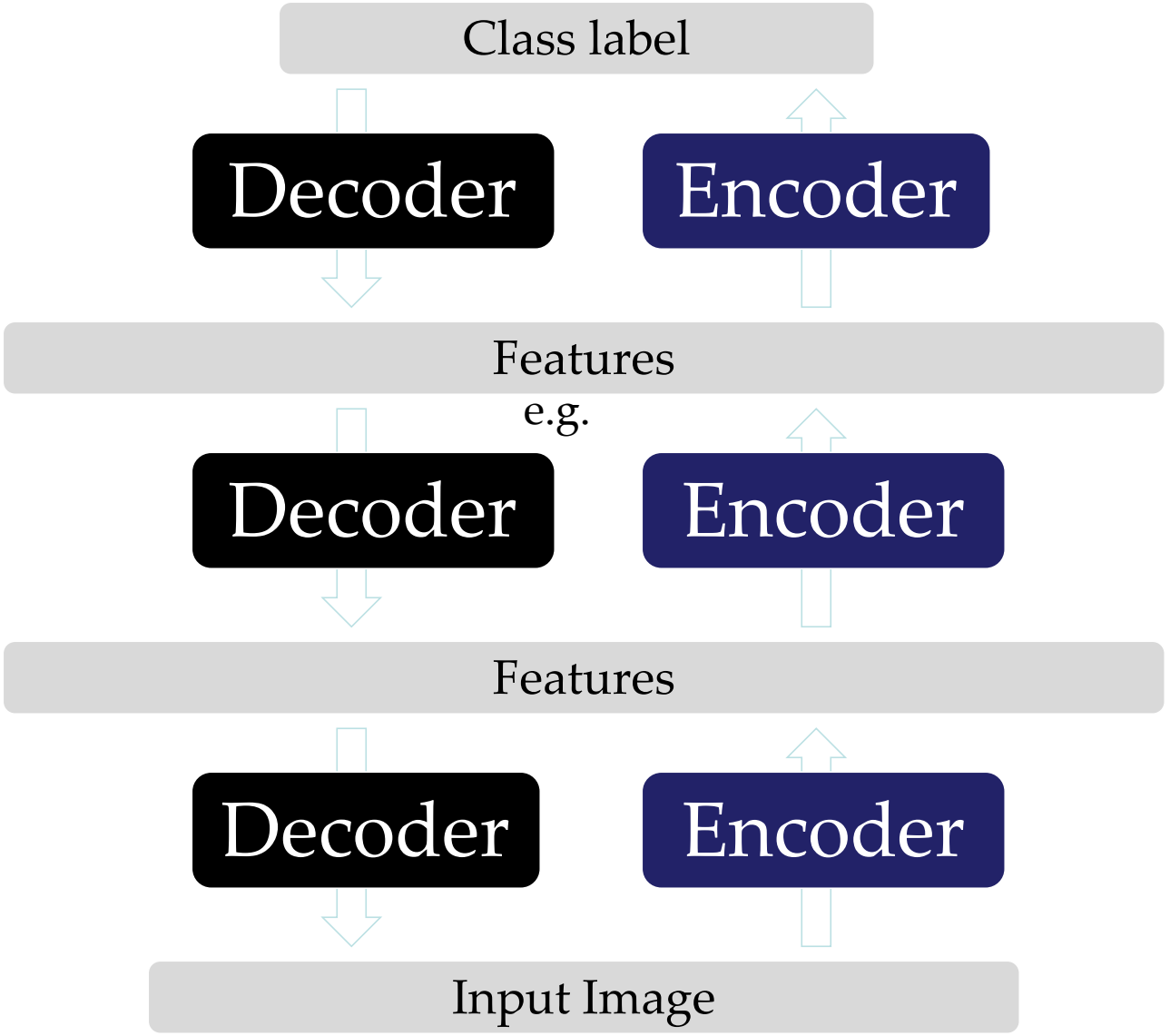
1）wake阶段：

认知过程，通过下层的输入特征(Input)和向上的认知(Encoder)权重产生每一层的抽象表示(Code)，再通过当前的生成(Decoder)权重产生一个重建信息(Reconstruction)，计算输入特征和重建信息残差，使用梯度下降修改层间的下行生成(Decoder)权重。也就是“如果现实跟我想象的不一样，改变我的生成权重使得我想象的东西变得与现实一样”。

2）sleep阶段：

生成过程，通过上层概念(Code)和向下的生成(Decoder)权重，生成下层的状态，再利用认知(Encoder)权重产生一个抽象景象。利用初始上层概念和新建抽象景象的残差，利用梯度下降修改层间向上的认知(Encoder)权重。也就是“如果梦中的景象不是我脑中的相应概念，改变我的认知权重使得这种景象在我看来就是这个概念”。

7.2 深度学习训练过程

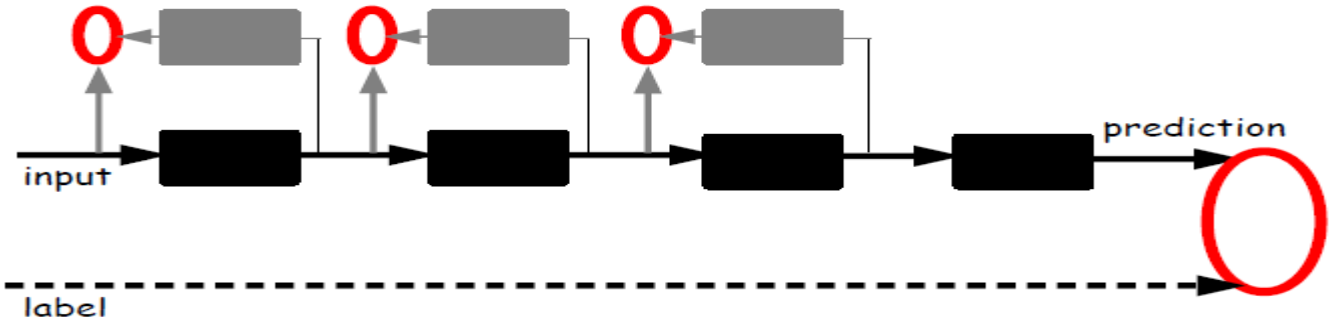

Class label
Decoder
Encoder
Features
e.g.
Decoder
Encoder
Features
Decoder
Encoder
Input Image

7.2 深度学习训练过程

- 第二步：自顶向下的监督学习

这一步是在第一步学习获得各层参数的基础上，在最顶的编码层添加一个分类器（例如Logistic回归、ANN、SVM等），然后通过带标签数据的有监督学习，利用梯度下降法去微调整个网络参数。

深度学习的第一步实质上是一个网络参数初始化过程。区别于传统神经网络初值随机初始化，深度学习模型是通过无监督学习输入数据的结构得到的，因而这个初值更接近全局最优，从而能够取得更好的效果。



Review:

- **Deep Learning:** a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for unsupervised feature learning and for pattern analysis/classification. The essence of deep learning is to compute hierarchical features or representations of the observational data, where the higher-level features or factors are defined from lower-level ones. (机器学习的一类技术，它通过分层结构的分阶段信息处理来探索无监督的特征学习和模式分析/分类。深度学习的本质是计算观察数据的分层特征或表示，其中高层特征或因子由低层特征得到。)

7.3 深度学习的常用模型及方法

- 自动编码器AutoEncoder（√）
- 稀疏自动编码器Sparse AutoEncoder
- 限制波尔兹曼机 Restricted Boltzmann Machine(RBM)
- 深度置信网络 Deep Belief Networks
- 卷积神经网络 Convolutional Neural Networks（√）

7.3 深度学习的常用模型及方法

- **自动编码器AutoEncoder**

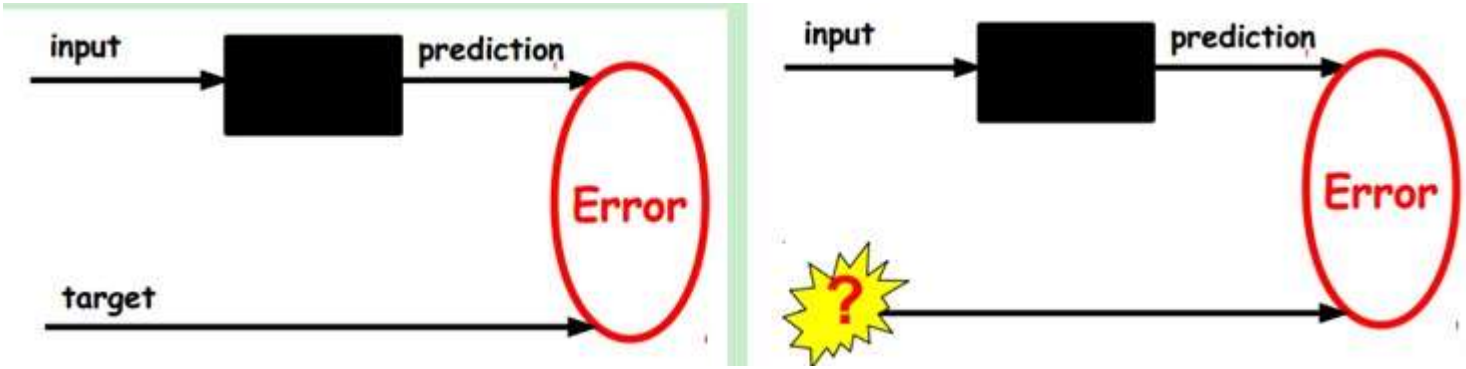
Deep Learning最简单的一种方法是利用人工神经网络(ANN)的特点，ANN本身就是具有层次结构的系统，若给定一个神经网络，我们假设其输出与输入是相同的，然后训练调整其参数，得到每一层中的权重。自然地，我们就得到了**输入I的几种不同表示**（每一层代表一种表示），这些表示就是特征。**自动编码器就是一种尽可能复现输入信号的神经网络**。为了实现这种复现，自动编码器必须捕捉能代表输入数据的最重要的因素，就像PCA那样，找到可以代表原信息的主要成分。

7.3 深度学习的常用模型及方法

- **自动编码器AutoEncoder**

  AutoEncoder具体过程简单说明如下：

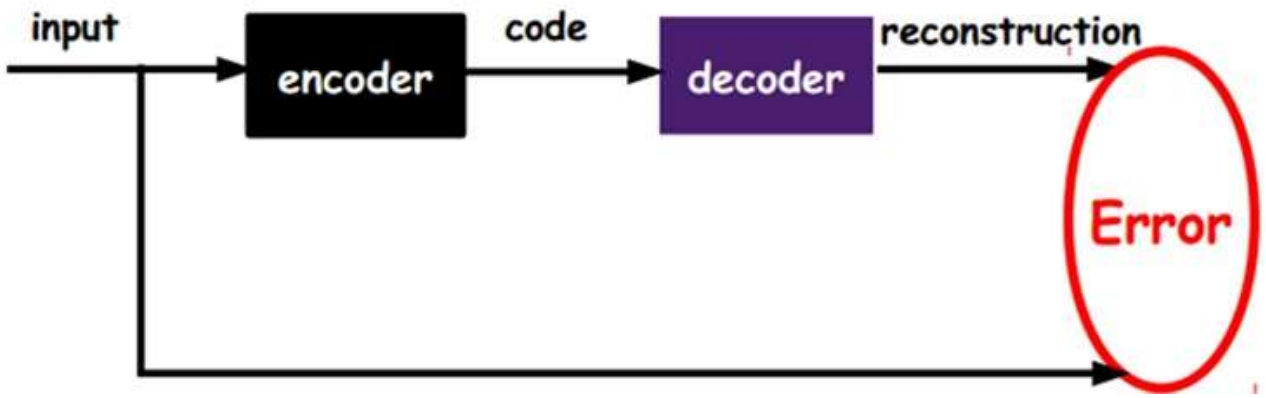
1) 给定无标签数据，用无监督学习方法学习特征



在之前的人工神经网络中，输入样本是有标签的(见左图)，即(input, target)，这样我们根据当前输出和target (label)之间的差去改变前面各层的参数，直到收敛。但现在我们只有无标签数据(见右图)，那么该误差如何得到呢？

7.3 深度学习的常用模型及方法

- **自动编码器AutoEncoder**



见上图，我们将input输入到encoder编码器，就会得到一个code，该code也就是输入的一个表示，那么如何知道该code表示的就是input呢？我们加一个decoder解码器，这时decoder就会输出一个信息，那么若输出的这个信息和开始的输入信号input很像（理想情况下输出和输入相同），显然，我们有理由相信该code是靠谱的。所以，我们就**通过调整encoder和decoder的参数，使得重构误差最小**，这时我们就得到了输入input信号的第一个表示，也就是编码code。因为是无标签数据，所以误差的来源就是直接重构后的输出与原输入相比较得到。

7.3 深度学习的常用模型及方法

- **自动编码器AutoEncoder**

2) 通过编码器产生特征，然后训练下一层，逐层训练下去

上面得到了第一层的code，重构误差最小使我们相信该code就是原输入信号的良好表达了，或者牵强点说，它和原信号是一模一样的(表达不一样，反映的是同一个东西)。第二层和第一层的训练方式没有什么差别，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，并且得到第二层输入的code，也就是原输入信息的第二个表达了。其他层按同样的方法炮制即可(训练这一层，前面层的参数都是固定的，并且它们的decoder没用了，就都不需要了)。

7.3 深度学习的常用模型及方法

- **自动编码器AutoEncoder**

3) 有监督微调 Supervised fine-tune

经过前面的过程，就能得到很多层。至于需要多少层(或者深度需要多少，目前还没有一个科学的评价方法)需要实验调试。每一层都会得到原始输入的不同的表达。当然，我们觉得它越抽象越好，就像人的视觉系统一样。

至此，该AutoEncoder还不能用于分类数据，因为它还没有学习如何去连结一个输入和一个类。它只是学会了如何去重构或者复现它的输入而已。或者说，它只是学习获得了一个可以良好代表输入的特征，这个特征能最大程度地代表原输入信号。

为了能实现分类，我们可在AutoEncoder最顶端的编码层添加一个分类器(如：Logistic Regression、ANN、SVM等)，然后采用标准的ANN有监督训练方法(如梯度下降法)训练该分类器或整个系统。

7.3 深度学习的常用模型及方法

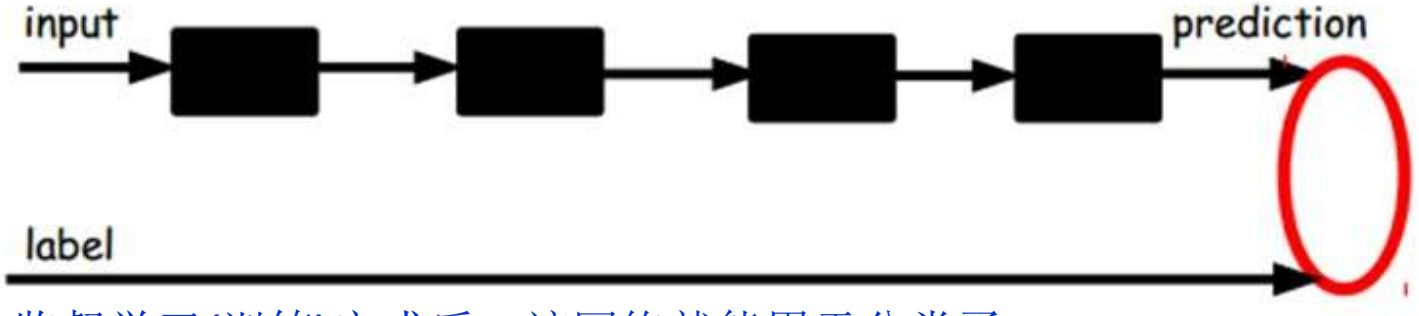
- 自动编码器AutoEncoder

3)有监督微调 Supervised fine-tune

就是说，我们需要将最后层的特征code输入到最后的分类器，通过有标签样本，采用有监督学习方法进行微调。微调方法可分两种：

一种微调是只调整分类器（见下图黑色部分）：

另一种微调是通过有标签的样本，微调整个系统（如果有足够多的数据样本，这种微调方式是最好的，end-to-end learning端对端学习）。

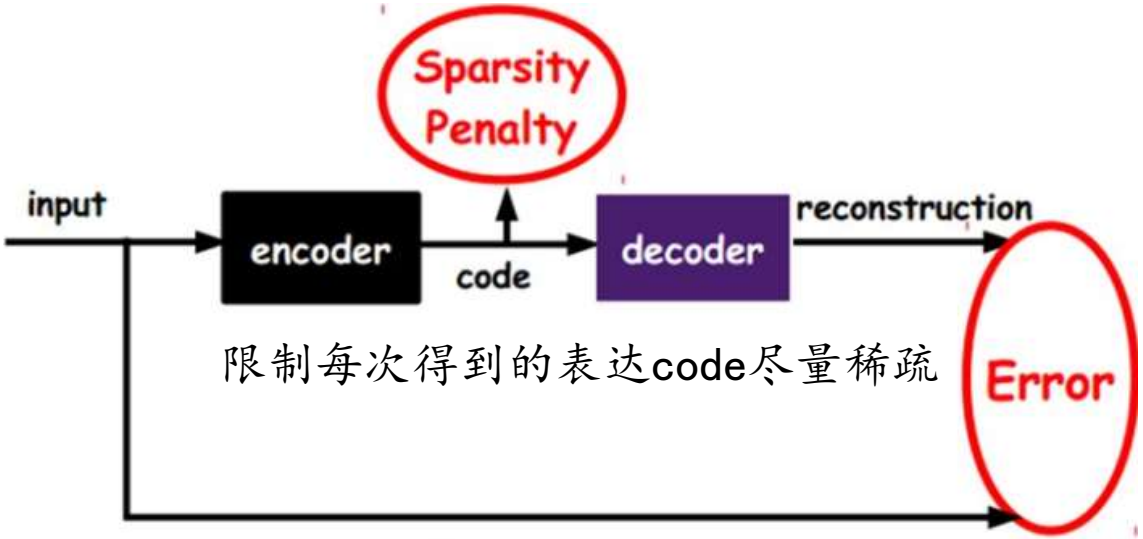


监督学习(训练)完成后，该网络就能用于分类了。

7.3 深度学习的常用模型及方法

- 稀疏自动编码器(Sparse AutoEncoder)

是AutoEncoder的一种改进变种：在AutoEncoder基础上加上一些约束条件得到新的Deep Learning方法。如，若在AutoEncoder的基础上加上L1的Regularity限制（L1主要是约束每一层中的节点中大部分都为0，只有少数不为0，这就是Sparse名称的由来），就可得到Sparse AutoEncoder方法。



- input: X code: $h = W^T X$

- loss: $L(X;W) = \|Wh - X\|^2 + \lambda \sum_j |h_j|$

见左图，其实就是限制每次得到的表达code尽量稀疏。因为稀疏的表达往往比其他的表达要有效(人脑似乎也是如此，某个输入只是刺激某些神经元，其他的大部分神经元是受到抑制的）。

7.3 深度学习的常用模型及方法

- 稀疏自动编码器(Sparse AutoEncoder)

1) **Training阶段**：给定一系列的样本图片[x1，x 2，…]，我们需要学习得到一组基[Φ1，Φ2，…]，也就是字典。

$$\min_{a,\phi} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$

可使用K-SVD方法交替迭代调整a [k]，Φ [k]，直至收敛，从而可以获得一组可以良好表示这一系列x的字典。

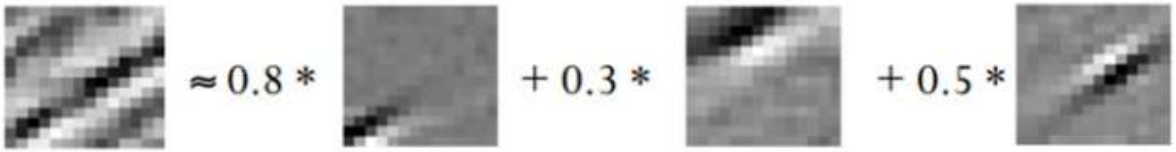
7.3 深度学习的常用模型及方法

- 稀疏自动编码器(Sparse AutoEncoder)

2) **Coding阶段**：给定一个新的图片x，由上面得到的字典，利用OMP算法求解一个LASSO问题得到稀疏向量a。这个稀疏向量就是这个输入向量x的一个稀疏表达。

$$\min_{a} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$

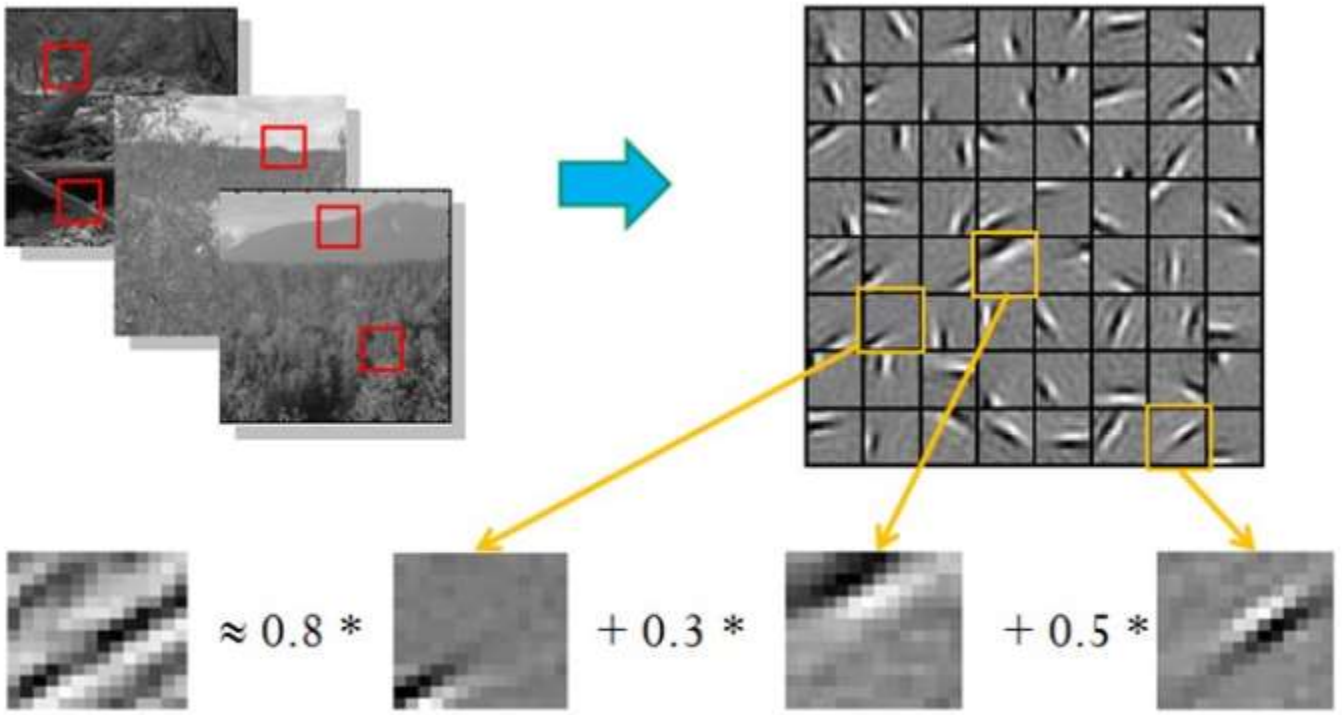
例如：



Represent x_i as: a_i = [0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]

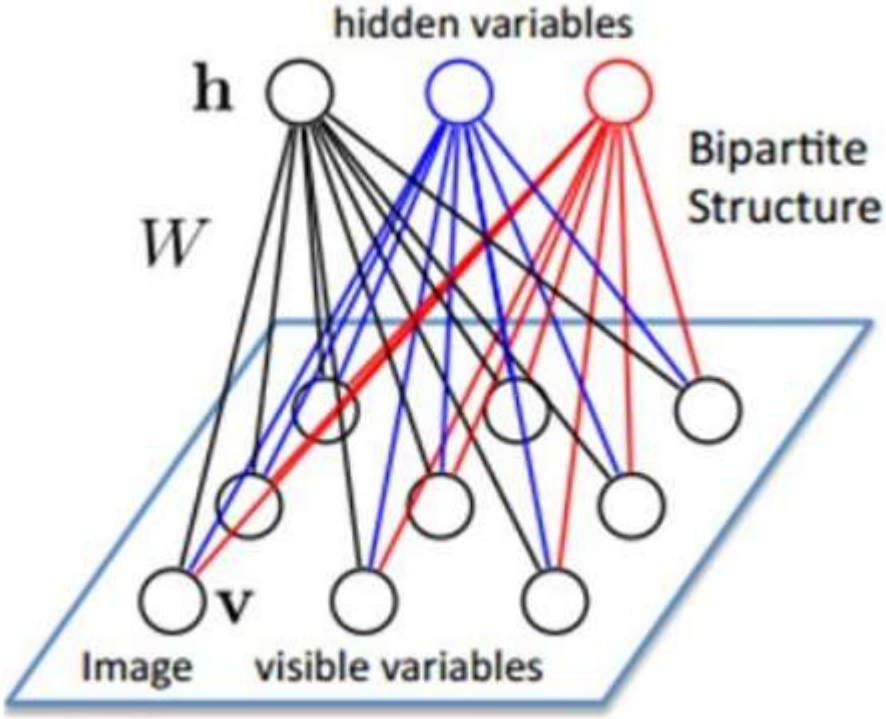
7.3 深度学习的常用模型及方法

- 稀疏自动编码器(Sparse AutoEncoder)



7.3 深度学习的常用模型及方法

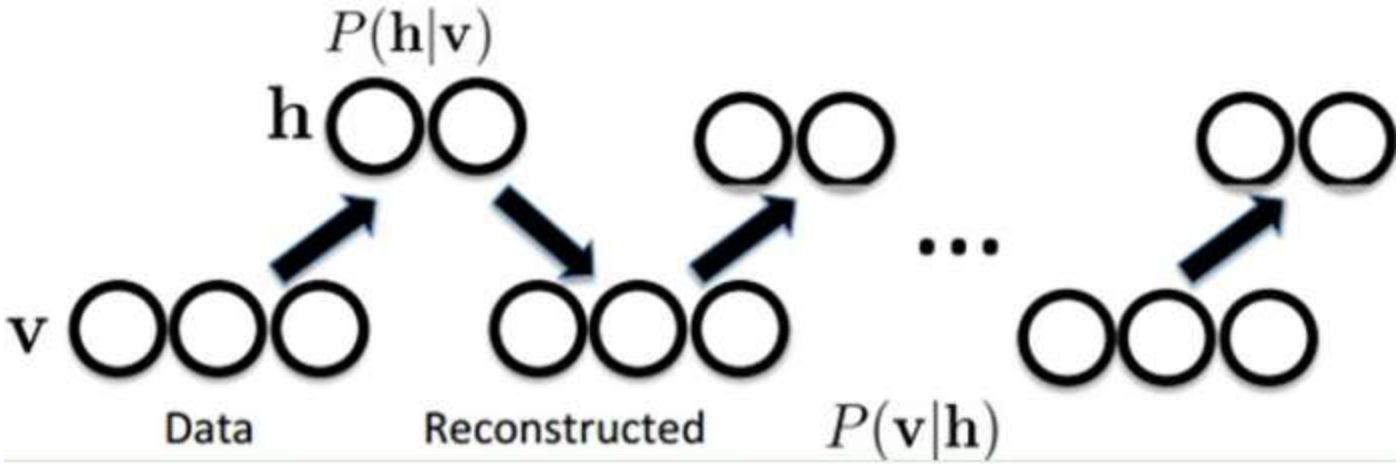
- 限制波尔兹曼机（Restricted Boltzmann Machine）



- 定义：假设有一个二部图，同层节点之间没有链接，一层是可视层，即输入数据层（v），一层是隐藏层(h)，如果假设所有的节点都是随机二值（ 0，1值）变量节点，同时假设全概率分布p(v,h)满足Boltzmann 分布，我们称这个模型是Restricted BoltzmannMachine (RBM)。

7.3 深度学习的常用模型及方法

- 限制波尔兹曼机（Restricted Boltzmann Machine）



- 限制波尔兹曼机（RBM）是一种深度学习模型。

7.3 深度学习的常用模型及方法

- 限制波尔兹曼机（Restricted Boltzmann Machine）

✓ 定义联合组态（joint configuration）能量：

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$\theta = \{W, a, b\}$ model parameters.

✓ 这样某个组态的联合概率分布可以通过Boltzmann 分布和这个组态的能量来确定：

$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right) = \underbrace{\frac{1}{\mathcal{Z}(\theta)}}_{\text{partition function}} \prod_{ij} \underbrace{e^{W_{ij} v_i h_j}}_{\text{potential functions}} \prod_i e^{b_i v_i} \prod_j e^{a_j h_j}$$

$$\mathcal{Z}(\theta) = \sum_{\mathbf{h},\mathbf{v}} \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right)$$

7.3 深度学习的常用模型及方法

- 限制波尔兹曼机（Restricted Boltzmann Machine）

✓ 给定隐层h的基础上，可视层的概率确定：

$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

（可视层节点之间是条件独立的）

✓ 给定可视层v的基础上，隐层的概率确定：

$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

7.3 深度学习的常用模型及方法

- 限制波尔兹曼机（Restricted Boltzmann Machine）

  待求问题：给定一个满足独立同分布的样本集：D={v(1), v(2),…, v(N)}，需要学习模型参数θ={W,a,b}。

  求解：

  最大似然估计：$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)}) - \frac{\lambda}{N}||W||_F^2$$

  我们需要选择一个参数，让我们当前的观测样本的概率最大

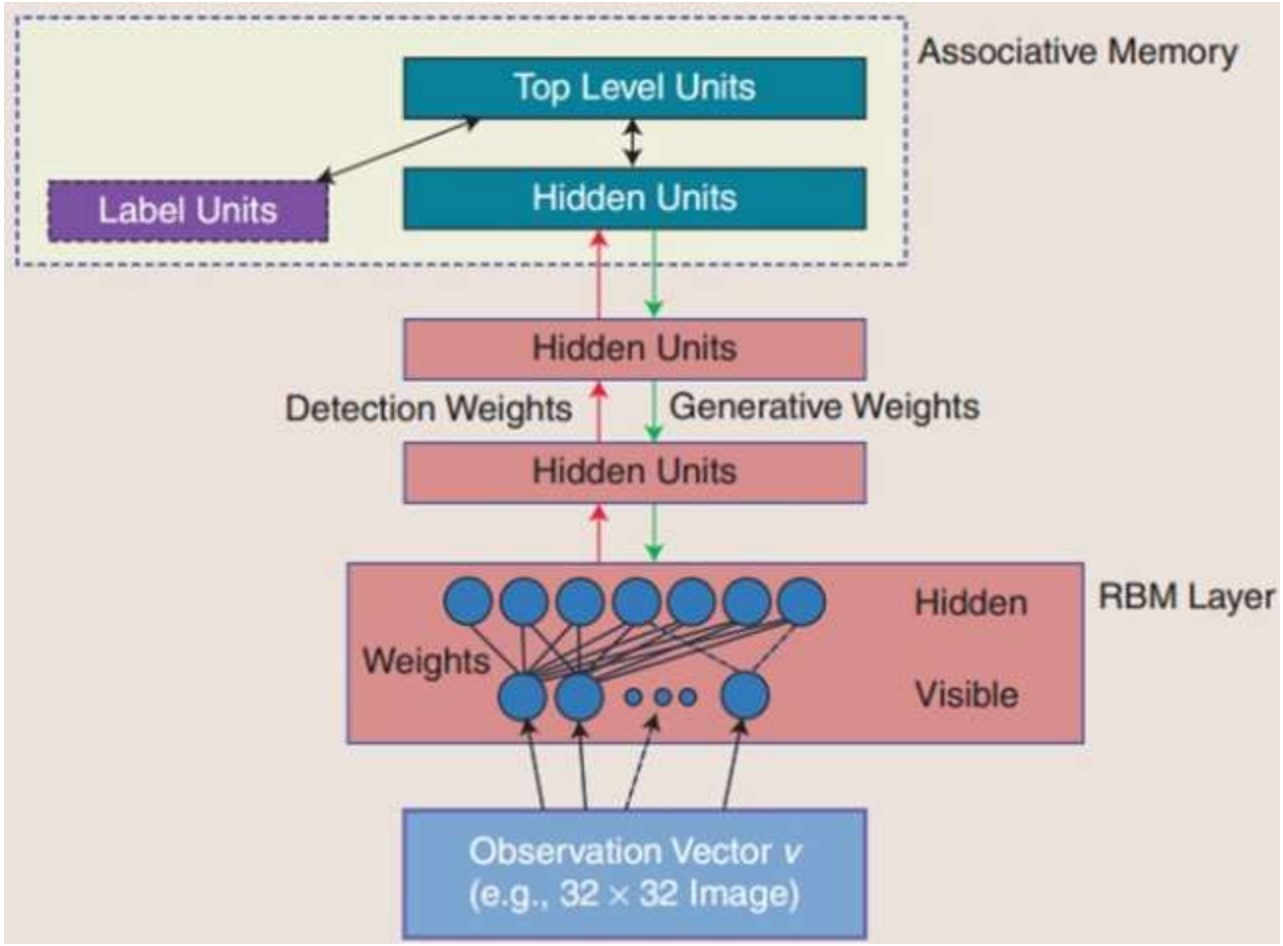
  对最大对数似然函数求导，即可得到L最大时对应的参数W：

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \mathrm{E}_{P_{data}}[v_i h_j] - \mathrm{E}_{P_\theta}[v_i h_j] - \frac{2\lambda}{N} W_{ij}$$

- 若隐藏层层数增加，可得到Deep Boltzmann Machine(DBM)

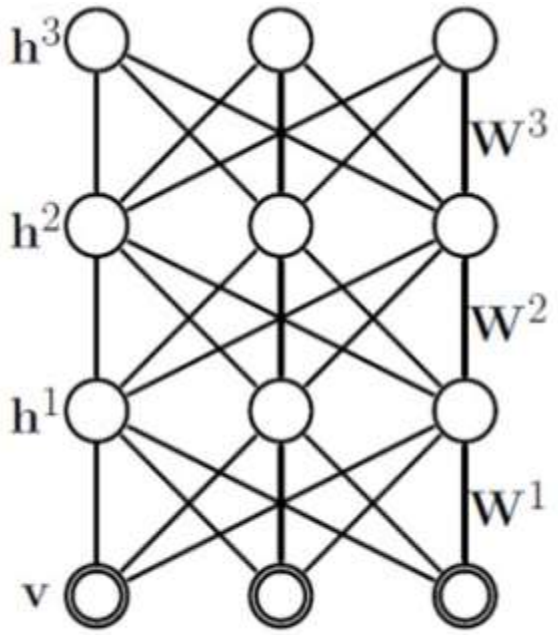
7.3 深度学习的常用模型及方法

- Deep Boltzmann Machine(DBM)

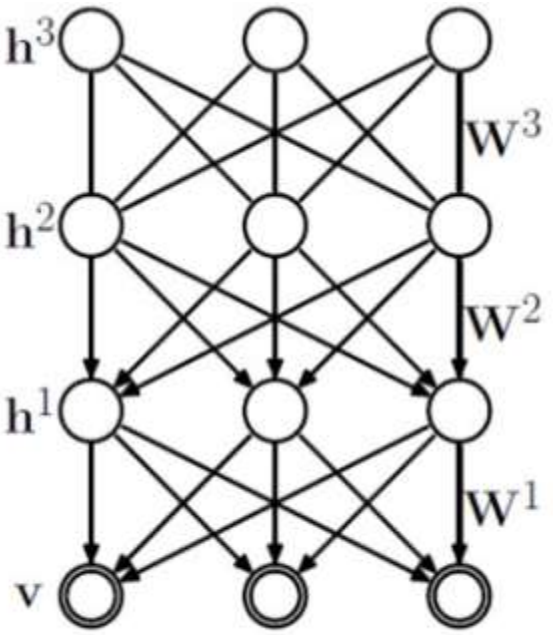


7.3 深度学习的常用模型及方法

- 深度置信网络（Deep Belief Networks）



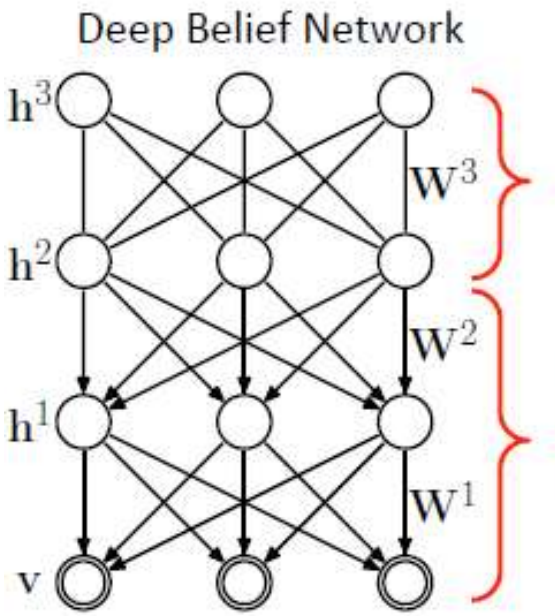
Deep Boltzmann Machine



Deep Belief Network

7.3 深度学习的常用模型及方法

- 深度置信网络（Deep Belief Networks）



The joint probability distribution factorizes:

$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = P(\mathbf{v}|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)P(\mathbf{h}^2, \mathbf{h}^3)$$

Sigmoid Belief Network　　RBM

$$P(\mathbf{h}^2, \mathbf{h}^3) = \frac{1}{\mathcal{Z}(W^3)} \exp\left[\mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

$$P(\mathbf{h}^1|\mathbf{h}^2) = \prod_j P(h_j^1|\mathbf{h}^2) \qquad P(h_j^1 = 1|\mathbf{h}^2) = \frac{1}{1 + \exp\left(-\sum_k W_{jk}^2 h_k^2\right)}$$

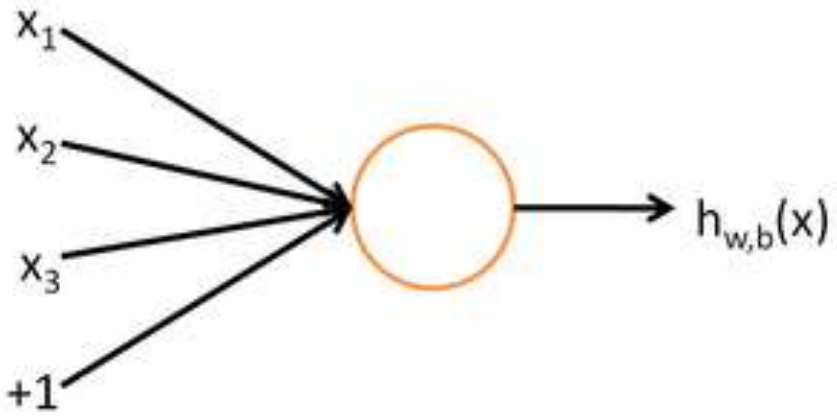
$$P(\mathbf{v}|\mathbf{h}^1) = \prod_i P(v_i|\mathbf{h}^1) \qquad P(v_i = 1|\mathbf{h}^1) = \frac{1}{1 + \exp\left(-\sum_j W_{ij}^1 h_j^1\right)}$$

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN (Convolutional Neural Networks)**

CNN是第一个被成功训练的多层深度神经网络结构，具有较强的容错、自学习及并行处理能力。最初是为识别二维图像而设计的多层感知器，局部连接和权值共享网络结构类似于生物神经网络。

1.神经网络回顾



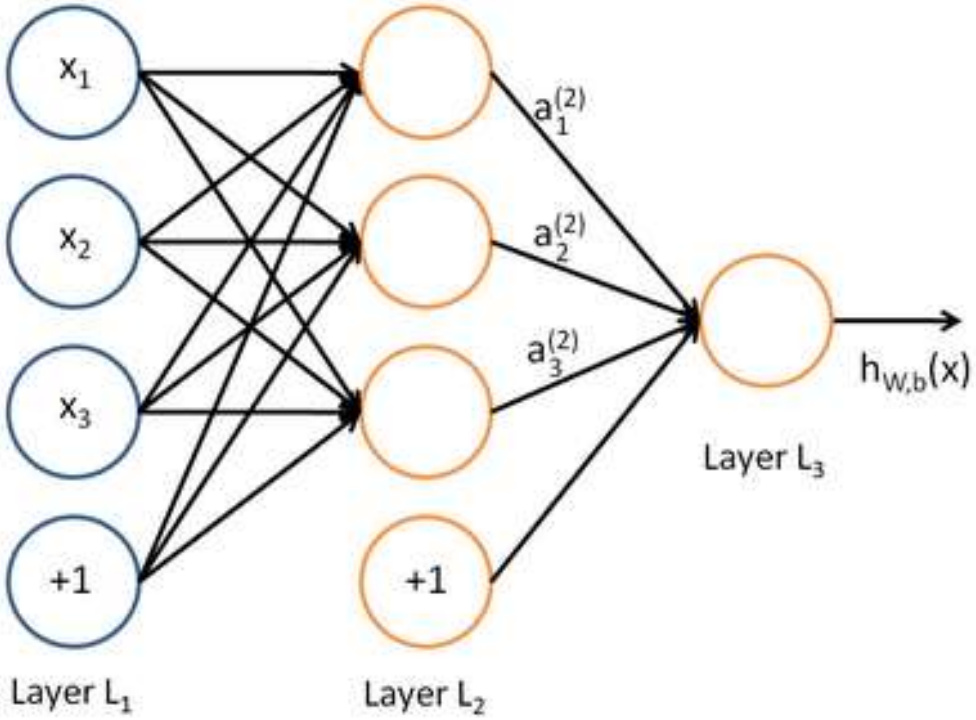
上图感知器单元，其对应公式为：

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

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

将多个感知器单元组合在一起并具有分层结构时，就形成了神经网络模型。下图展示了一个具有一个隐含层的神经网络。



7.3 深度学习的常用模型及方法

- **卷积神经网络CNN (Convolutional Neural Networks)**

其对应的公式为：

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

与此类似，可拓展到有2、3、4、5，…多个隐含层。

神经网络的训练方法与Logistic类似，不过由于其多层性，还需要利用链式求导法则对隐含层的节点进行求导，即“梯度下降+链式求导”法则，专业名称为反向传播Back propagation。关于神经网络的学习训练算法，见前面的BP神经网络(此略)。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN (Convolutional Neural Networks)**

2.卷积神经网络CNN

在图像处理中，图像可表示为像素的向量，如一张1000×1000的图片，可表示为1000000的向量。在上一节提到的神经网络中，若隐含层数目与输入层一样，即也是1000000时，那么输入层到隐含层的参数数据为$1000000\times1000000=10^{12}$，这么多的参数基本没法训练。因此，要想采用神经网络对输入图像进行处理识别，必须先减少参数加快速度。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

2.卷积神经网络CNN

卷积神经网络是深度神经网络中的一种，已成为当前语音分析和图像识别领域的研究热点。它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。该优点在网络的输入是多维图像时表现更为明显，使**图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程**。卷积网络最初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、倾斜或者其他形式的变形具有高度不变性。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

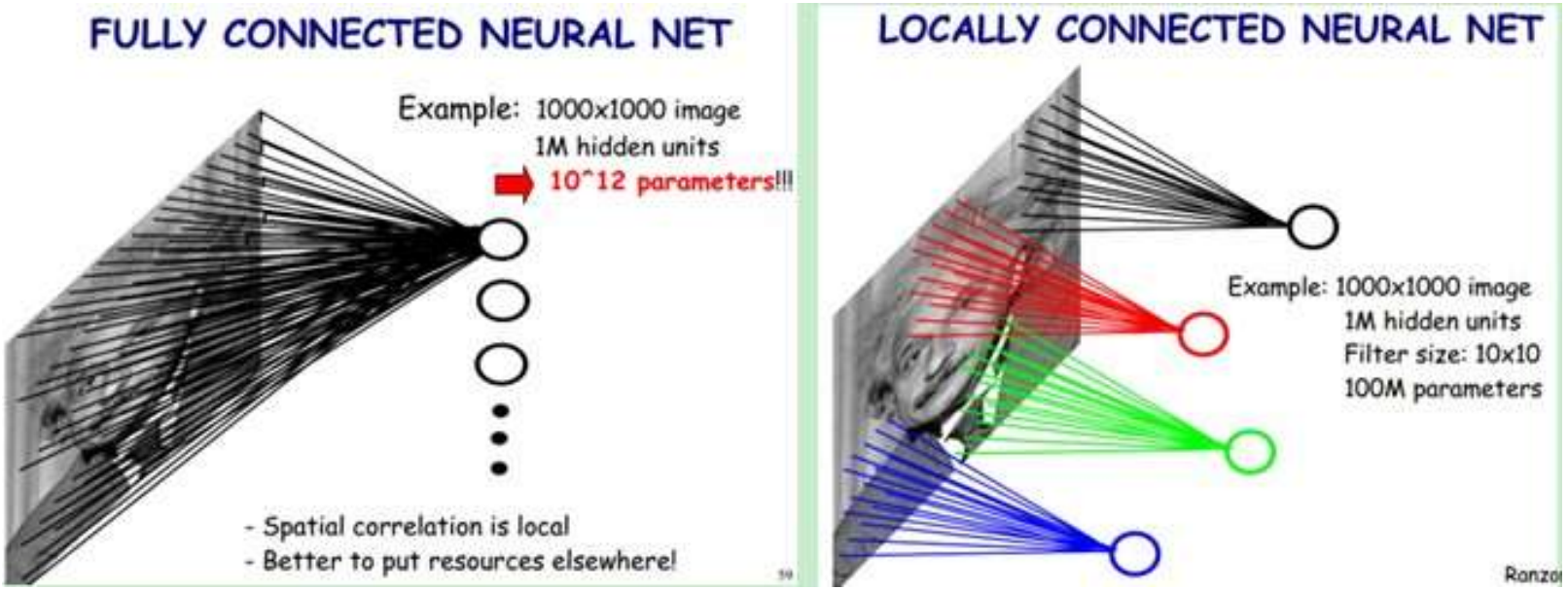
2.1 局部感知

卷积神经网络有两种神器可以减少参数数目，**第一种神器**叫做**局部感知域**。一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部信息综合起来就得到了全局的信息。网络部分连通的思想，也是受生物学中的视觉系统结构的启发。**视觉皮层的神经元就是局部接受信息的(即这些神经元只响应某些特定区域的刺激)**。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

如下图所示，左图为全连接，右图为局部连接。



在右图中，假设每个神经元只和10×10个像素值相连，那么权值数据为1000000×100个参数，减少为原来的万分之一。而那10×10个像素值所对应的10×10个参数，其实就相当于卷积操作。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

2.2 参数共享

用局部感知域参数仍然较多，于是启动**第二种神器**，称作**权值共享**。在上面右图的局部连接中，每个神经元都对应100个参数，一共1000000个神经元，如果这1000000个神经元的100个参数都是相等的，那么参数数目就变为100个了。

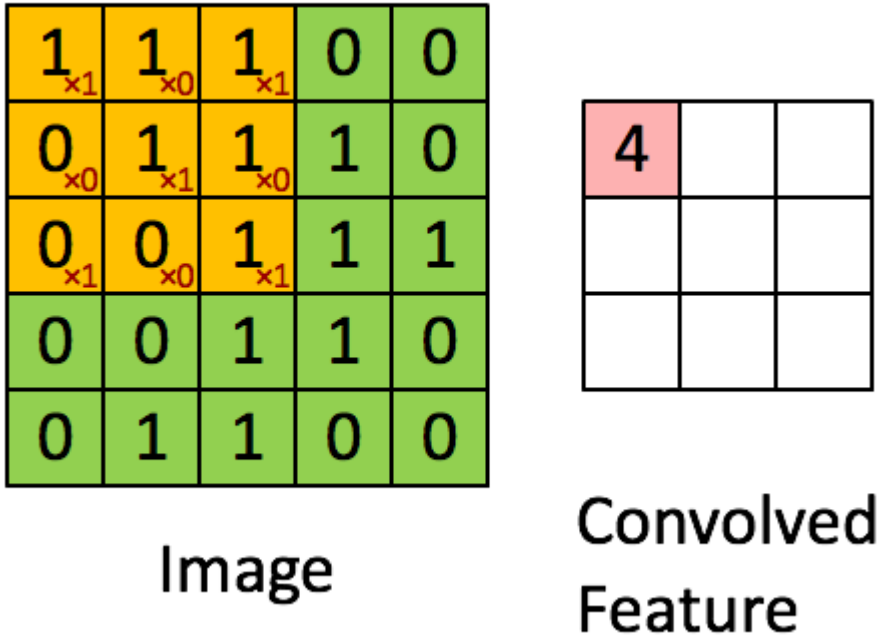
如何理解权值共享？我们可以将这100个参数（即卷积操作）看成是特征提取的方式，该方式与位置无关。这其中隐含的基本原理是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。更直观一些，当从一个大尺寸图像中随机选取一小块(patch)，比如说 8*8 作为样本，并且从这个小块样本中学习到了一些特征，这时我们能把从这个 8*8 样本中学习到的特征作为探测器，应用到这个图像的任意地方中去。特别地，我们能用从 8*8 样本中所学习到的特征跟原本的大尺寸图像作卷积，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

2.2 参数共享

下图展示了一个3×3的*卷积核*在5×5的图像上做卷积的过程。每个卷积都是一种特征提取方式，就像一个筛子，将图像中符合条件（激活值越大越符合条件）的部分筛选出来。

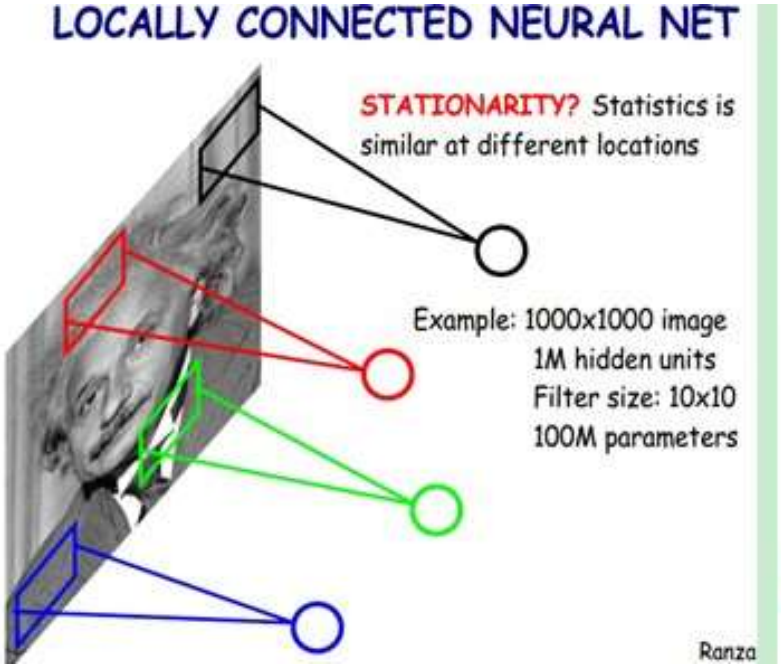


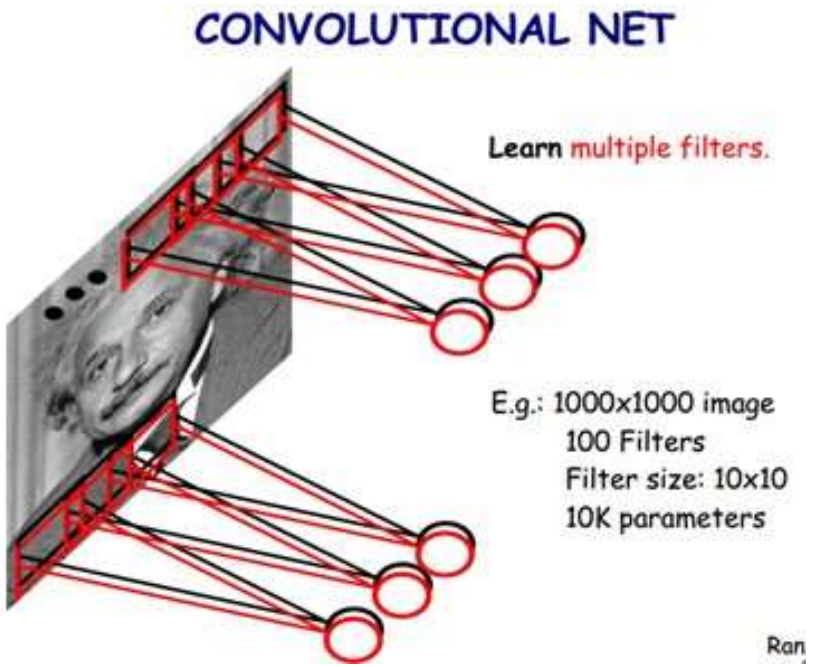
7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

2.3 多核卷积

上面所述只有100个参数时，表明只有1个10×10的卷积核，显然，特征提取是不充分的，我们可以添加多个卷积核，比如32个卷积核，可以学习32种特征。当有多个卷积核时，见下图所示。





在右图中，不同颜色表示不同的卷积核。每个卷积核都会将一幅图像生成为另一幅图像。比如两个卷积核就可以将其生成两幅图像，这两幅图像可以看做是一张图像的不同通道。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

  2.4 Down-pooling

  在通过卷积运算获得了特征 (features) 之后，下一步我们希望利用这些特征去做分类。理论上讲，人们能用所有提取得到的特征去训练分类器，但这样做面临计算量的挑战。例如：对于一个 96×96 像素的图像，假设我们已经学习得到了400个定义在8×8输入上的特征，每一个特征和图像卷积都会得到一个 $(96-8+1)\times(96-8+1)=7921$ 维的卷积特征，由于有 400 个特征，所以每个样例 (example) 都会得到一个 7921×400 = 3,168,400 维的卷积特征向量。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)。

7.3 深度学习的常用模型及方法

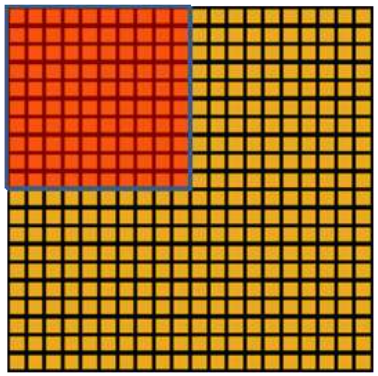
- **卷积神经网络CNN(Convolutional Neural Networks)**

为了避免对高维特征进行分类造成的over-fitting问题，我们先回顾一下，之所以决定使用卷积后的特征是因为**图像具有**一种“静态性(stationarity)”的**属性**，这就意味着在一个图像区域有用的特征极有可能在另一个区域同样适用。因此，为了描述大的图像，很自然的一个想法就是对不同位置的特征进行**聚合统计**，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(不易过拟合)。这种聚合操作被称为**池化** (pooling)，有时也称为平均池化或者最大池化等 (取决于计算池化的方法)。

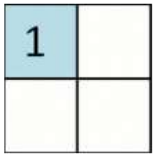
7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

下图展示了对卷积特征的池化 (pooling)过程：



Convolved feature



Pooled feature

点评：卷积神经网络的核心思想是将局部感受域、权值共享(或者权值复制)以及时间或空间亚采样这三种结构思想巧妙地结合起来获得了某种程度的位移、尺度、形变不变性。 这种综合创新思维方法值得我们学习借鉴。

2.5 多层卷积

在实际应用中，往往使用多层卷积，然后再使用全连接层进行训练，多层卷积的目的是一层卷积学到的特征往往是局部的，层数越高，学到的特征就越全局化。

至此，卷积神经网络的基本结构和原理阐述完毕。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

3. CNN训练算法

CNN训练算法与传统的BP算法差不多。主要包括4步，这4步被分为两个阶段：

第1阶段，向前传播阶段：

a）从样本集中取一个样本(X,Yp)，将X输入网络；

b）计算相应的实际输出Op。

在此阶段，信息从输入层经过逐级的变换，传送到输出层。这个过程也是网络在完成训练后正常运行时执行的过程。在此过程中，网络执行的是计算（实际上就是输入与每层的权值矩阵相点乘，得到最后的输出结果）：

Op=Fn（…（F2（F1（XpW（1））W（2））…）W（n））

第2阶段，向后传播阶段：

a）算实际输出Op与相应的理想输出Yp的差；

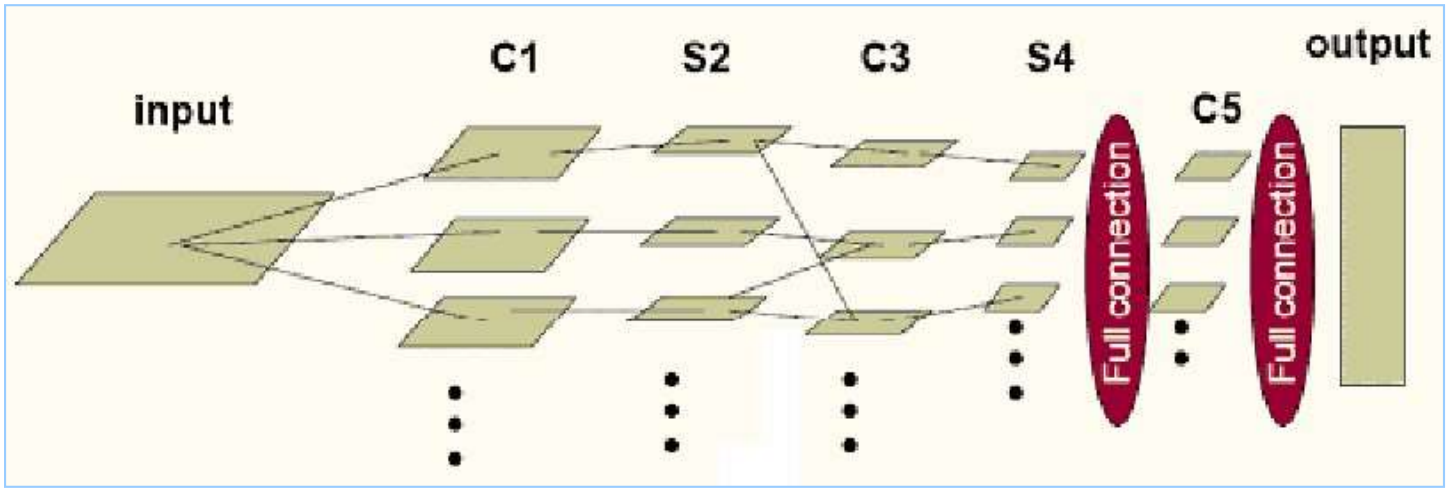
b）按极小化误差方法反向传播调整权值矩阵。

7.3 深度学习的常用模型及方法

- 典型的卷积神经网络结构原理小结：

  如下图所示。CNN是一种多层前向网络，每层由多个二维平面组成，每个平面由多个神经元组成。

  网络输入为二维视觉模式；作为网络中间层的卷积层(Convolutional Layer, C)和子采样层(Subsampling Layer, S)交替出现；网络输出层为前馈网络的全连接方式，输出层的维数为分类任务的类别数。



卷积神经网络结构图

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

4. LeNet-5网络结构(经典的CNN设计，用于复杂手写数字的识别)

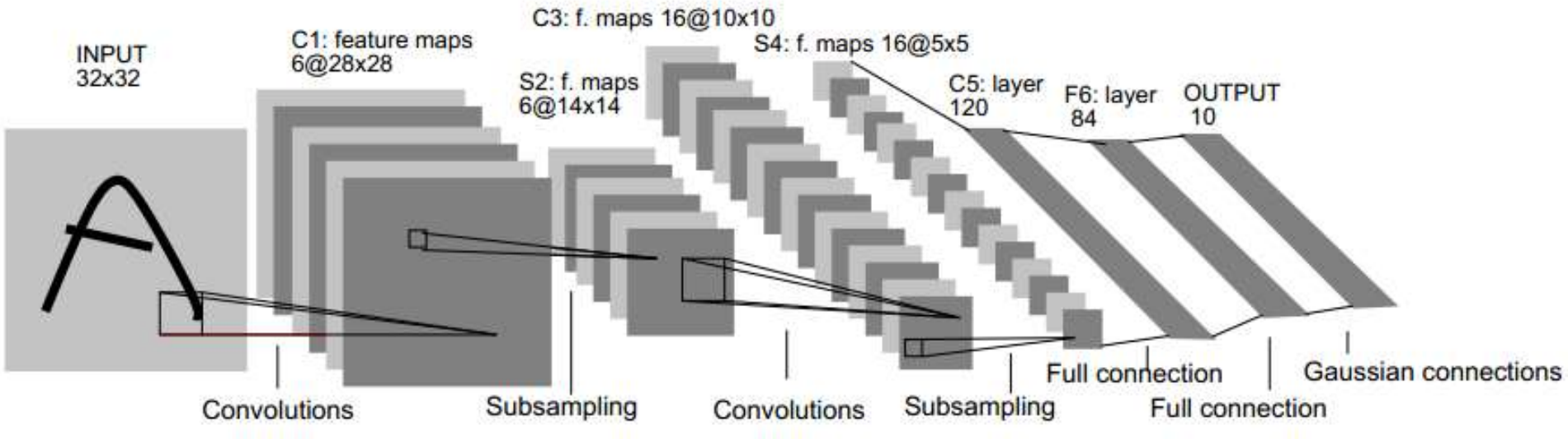
基础知识回顾：convolution和pooling的优势为使网络结构中所需学习到的参数数目变得更少，并且学习到的特征具有一些不变性，比如说平移、旋转不变性。以2D图像提取为例，学习的参数个数变少是因为不需要用整张图片的像素输入到网络，而只需学习其中一部分patch。而不变的特性则是由于采用了mean-pooling或者max-pooling等方法。

LeNet-5是美国贝尔实验室著名学者Yann LeCun(美籍法国人) 采用CNN开发的手写字符分类系统，在商业上取得了极大的成功。

7.3 深度学习的常用模型及方法

- 卷积神经网络CNN(Convolutional Neural Networks)

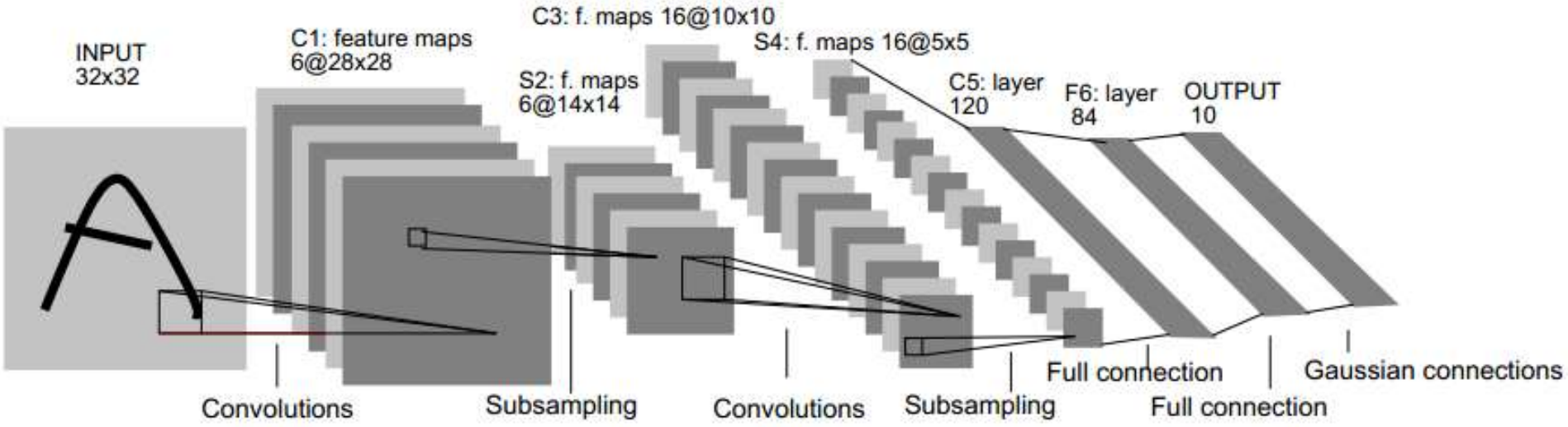
4. LeNet-5网络结构(经典的CNN设计，用于复杂手写数字的识别)



(1)输入层：32*32大小的图片。每输入一张32*32大小的图片，就输出一个84维的向量(F6：Full connection)，该向量就是我们提取出的特征向量。

(2)第一隐层(C1卷积层)：卷积核的数目6，卷积核大小(即局部感受域的大小)为5*5，输入层经 C1卷积层，输入图像被卷积成6个28*28的特征图，特征图的大小由输入特征图的32*32变成输出的28*28，卷积采用VALID方式，即输出特征图大小=输入特征图大小-(卷积核大小-1)=32-(5-1)，其中每次移动步长为1个像素。

7.3 深度学习的常用模型及方法



(3) 第二隐层(S2抽样层)：其局部感受域大小为2×2， 每个2×2的像素被下采样为1个像素(即每次对2×2的4个像素进行pooling得到1个值)，s2层变成了6张14×14大小的特征图。

(4)第三隐层(C3卷积层)：卷积核的数目为16个，卷积核的大小为5×5，因此经过该层输出的特征图大小为10×10(10=14-5+1)。需要注意的是，该层输入的6个特征图变成输出的16个特征图，这个过程可看作是将S2的特征图用1个输入层为150(5×5×6，而不是5×5)个结点、输出层为16个结点的网络进行convolution 。并且C3的每个特征图并不是和S2的每个特征图都相连，而是可能只和其中几个进行连接。

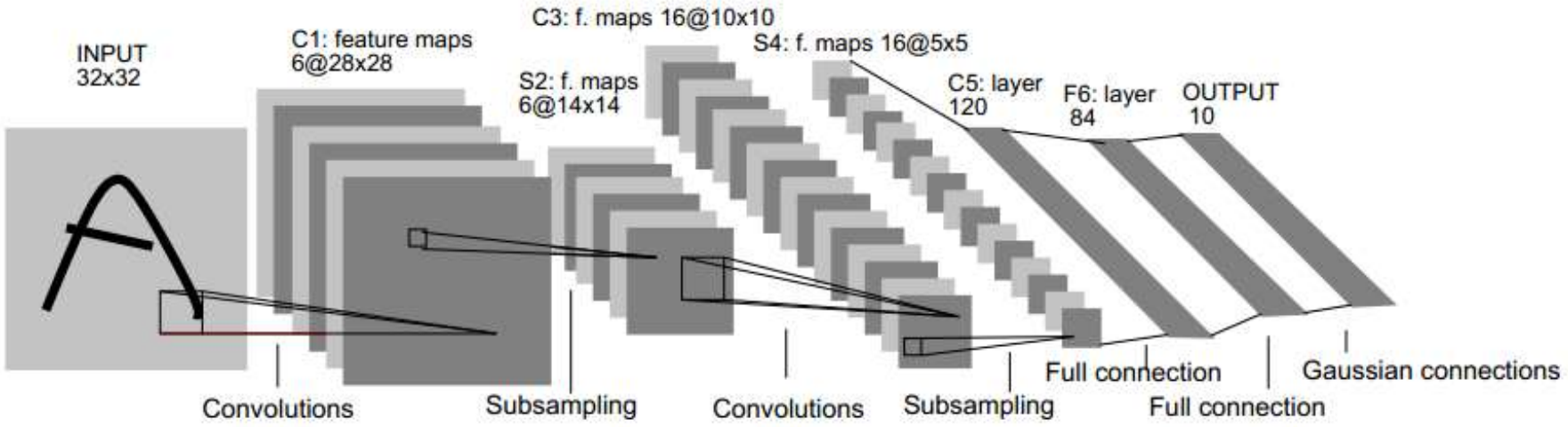
7.3 深度学习的常用模型及方法

LeNet-5的S2层和C3层具体连接关系见下图，纵坐标表示输入特征图索引，横坐标表示输出特征图索引，有X标记的位置表示该位置对应的输入特征图与输出特征图之间存在连接。

	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

图　LeNet-5中S2和C3层特征图的连接关系

7.3 深度学习的常用模型及方法



(5) 第四隐层(S4下采样层)：其局部感受域大小为2×2， 每个2×2的像素被下采样为1个像素(即每次对2×2的4个像素进行pooling得到1个值)，s4层变成了16个5×5大小的特征图。

(6)第五隐层(C5卷积层)：卷积核的数目为6个，卷积核的大小为5×5，故C5特征图的大小为1×1，在卷积操作完成后，C5将得到的特征图展开成一个向量，向量大小为120(=20*6)。

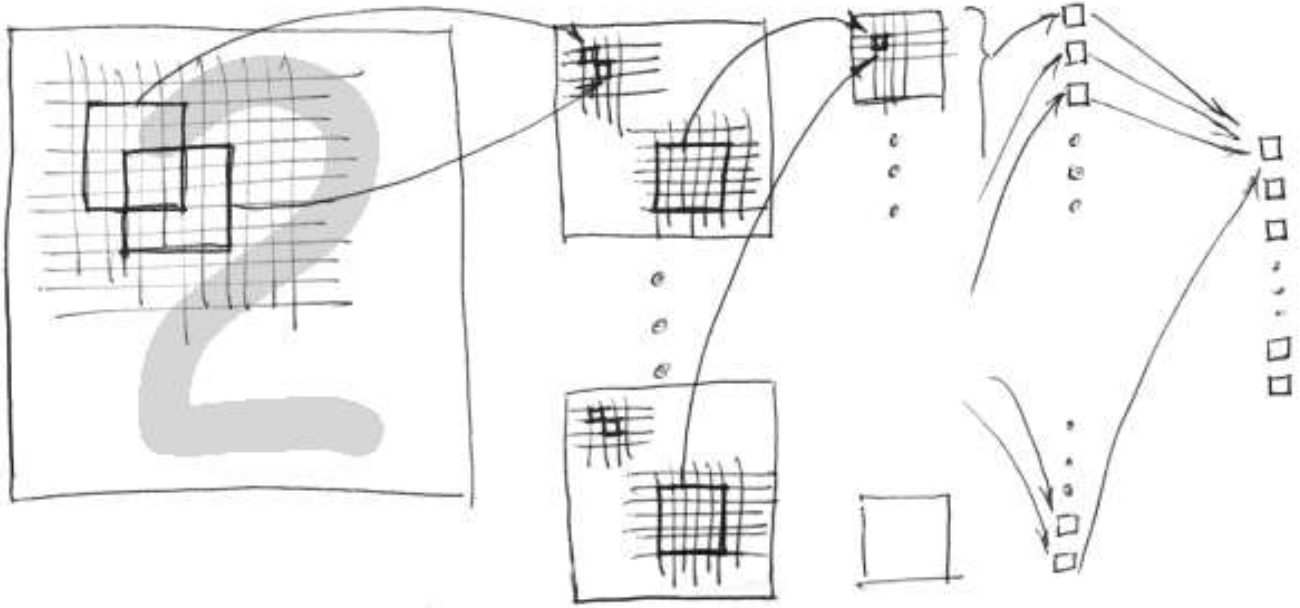
(7)第六隐层(F6全连接网络)：F6为一个全连接网络，结点数为84，这84个结点与C5的120个输入结点完全相连。

(8)输出层：输出层结点数目为10，表示对应问题的分类数目(0~9共10个类别)。

7.3 深度学习的常用模型及方法

例：一种简化的LeNet-5 CNN系统。

简化的LeNet-5系统把下采样层和卷积层结合起来，避免了下采样层过多的参数学习过程，同样保留了对图像位移、扭曲的鲁棒性，其网络结构图如下。



7.3 深度学习的常用模型及方法

简化的LeNet-5系统包括输入层的话，只有5层结构，而经典的LeNet-5结构不包含输入层就已经是7层网络结构了。它实现下采样非常简单，直接取其第一个位置节点上的值即可。

1、**输入层：**MNIST手写数字图像的大小是28×28的，这里通过补零扩展为29×29的大小。这样输入层神经节点个数为29×29等于841个。

2、**第一层：**由6张不同的特征映射图组成。每一张特征图的大小是13×13。注意，由于卷积窗大小为5×5，加上下采样过程，易得其大小为13×13。所以，第二层共有6×13×13等于1014个神经元节点。每一张特征图加上偏置共有5×5+1等于26个权值需要训练，总共有6×26等于156个不同的权值。即总共有1014×156=26364条连接线。

7.3 深度学习的常用模型及方法

3、**第二层：**由50张不同的特征映射图组成。每一张特征图的大小是5×5。注意，由于卷积窗大小为5×5，加上下采样过程，易得其大小为5×5。由于上一 层是由多个特征映射图组成，那么，如何组合这些特征形成下一层特征映射图的节点呢？简化的LeNet-5系统采用全部所有上层特征图的组合。也就是原始LeNet-5特征映射组合图的最后一列的组合方式。因此，总共有5×5×50等于1250 个神经元节点，有(5×5+1) ×6×50等于7800 个权值，总共有1250×26=32500条连接线。

4、**第三层：**这一层是一个一维线性排布的网络节点,与前一层是全连接的网络，其节点个数设为为100，故而总共有100×(1250+1)等于125100个不同的权值，同时，也有相同数目的连接线。

5、**第四层：**这一层是网络的输出层，如果要识别0-9数字的话，就是10个节点。该层与前一层是全连接的，故而，总共有10×(100+1)等于1010个权值，有相同数目的连接线。

7.3 深度学习的常用模型及方法

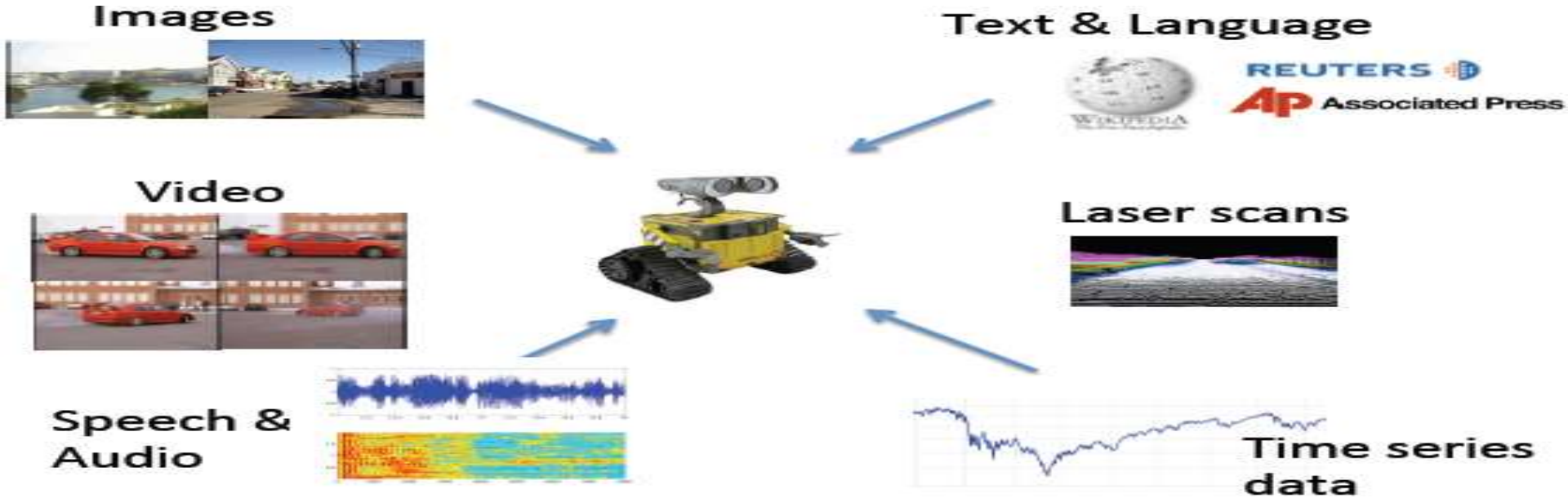
- 卷积神经网络CNN(Convolutional Neural Networks)

5. ImageNet-2010网络结构(采用CNN设计，NIPS2012)

ImageNet LSVRC是一个图片分类的国际竞赛，其训练集包括127W+张图片，验证集有5W张图片，测试集有15W张图片。这里截取2010年Alex Krizhevsky, Hinton G E. Imagenet classification with deep convolutional neural networks一文所设计的一种CNN网络结构，该结构在2010年取得冠军，top-5错误率为15.3%。值得一提的是，在ImageNet LSVRC2013竞赛中，取得冠军的GoogNet达到了top-5错误率6.67%。可见，深度学习的提升空间还很巨大。

7.3 深度学习的常用模型及方法

下图为Alex的CNN网络结构图。需要注意的是，该模型采用了2-GPU并行结构，即第1、2、4、5卷积层都是将模型参数分为2部分进行训练的。在这里，更进一步，并行结构分为数据并行与模型并行。数据并行是指在不同的GPU上，模型结构相同，但将训练数据进行切分，分别训练得到不同的模型，然后再将模型进行融合。而模型并行则是，将若干层的模型参数进行切分，不同的GPU上使用相同的数据进行训练，得到的结果直接连接作为下一层的输入。



7.3 深度学习的常用模型及方法

上图模型的基本参数如下：
输入：224×224大小的图片，3通道
第一层卷积：11×11大小的卷积核96个，每个GPU上48个。
第一层max-pooling：2×2的核。
第二层卷积：5×5卷积核256个，每个GPU上128个。
第二层max-pooling：2×2的核。
第三层卷积：与上一层是全连接，3*3的卷积核384个。分到两个GPU上个192个。
第四层卷积：3×3的卷积核384个，两个GPU各192个。该层与上一层连接没有经过pooling层。
第五层卷积：3×3的卷积核256个，两个GPU上个128个。
第五层max-pooling：2×2的核。
第一层全连接：4096维，将第五层max-pooling的输出连接成为一个一维向量，作为该层的输入。
第二层全连接：4096维
Softmax层：输出为1000，输出的每一维都是图片属于该类别的概率。

7.3 深度学习的常用模型及方法

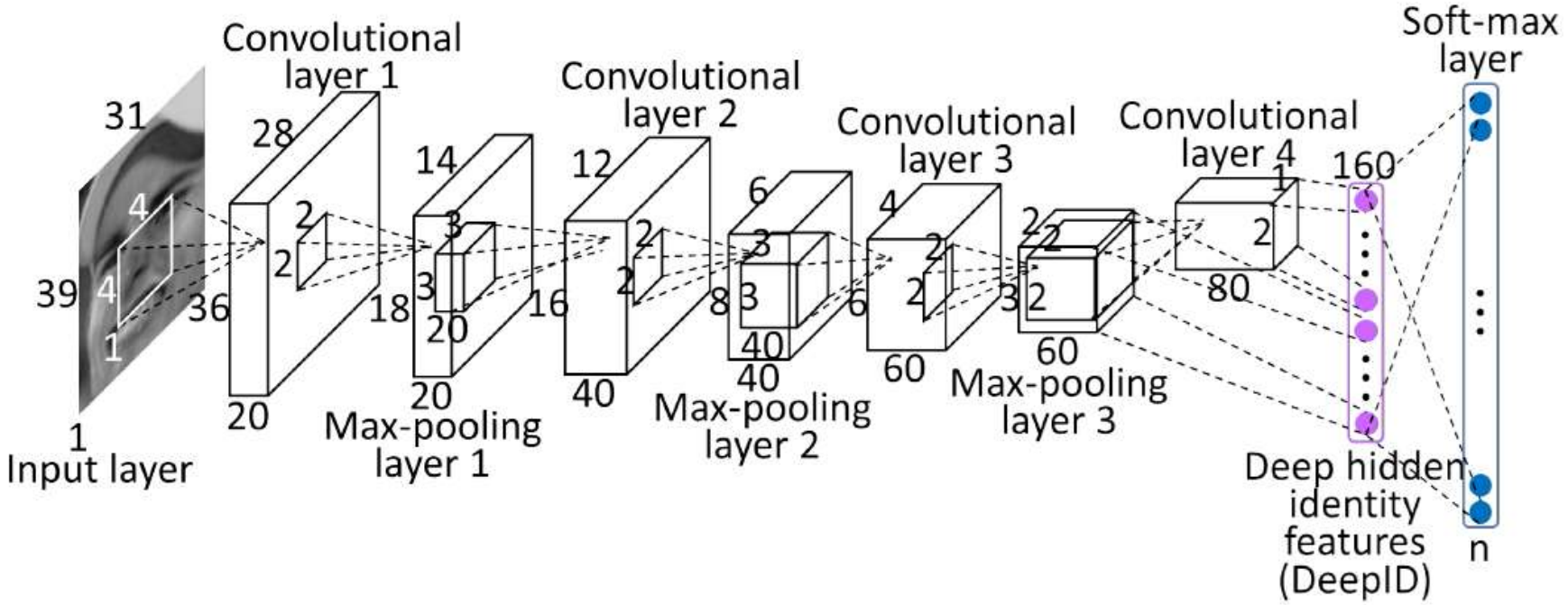
- 卷积神经网络CNN(Convolutional Neural Networks)

5. DeepID网络结构(采用CNN设计，CVPR2014)

DeepID CNN网络结构是香港中文大学的Sun Yi, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes(CVPR2014)开发出来的用于学习人脸特征的卷积神经网络。每张输入的人脸被表示为160维的向量，学习到的向量经其他模型进行分类，在人脸验证试验上得到了97.45%的正确率，更进一步，论文作者改进了CNN，又得到了99.15%的正确率，可见DeepID比人眼识别人脸更为厉害，被誉为为国际上第三代人脸识别技术。

7.3 深度学习的常用模型及方法

见下图，该结构与ImageNet的具体参数类似，这里只解释一下不同。



上图结构中，在最后只有一层全连接层，然后就是softmax层。论文中是以该全连接层作为图像的表示。在全连接层，以第四层卷积和第三层max-pooling的输出作为全连接层的输入，这样就能学习到局部和全局特征。

7.3 深度学习的常用模型及方法

- **卷积神经网络CNN(Convolutional Neural Networks)**

CNN总结

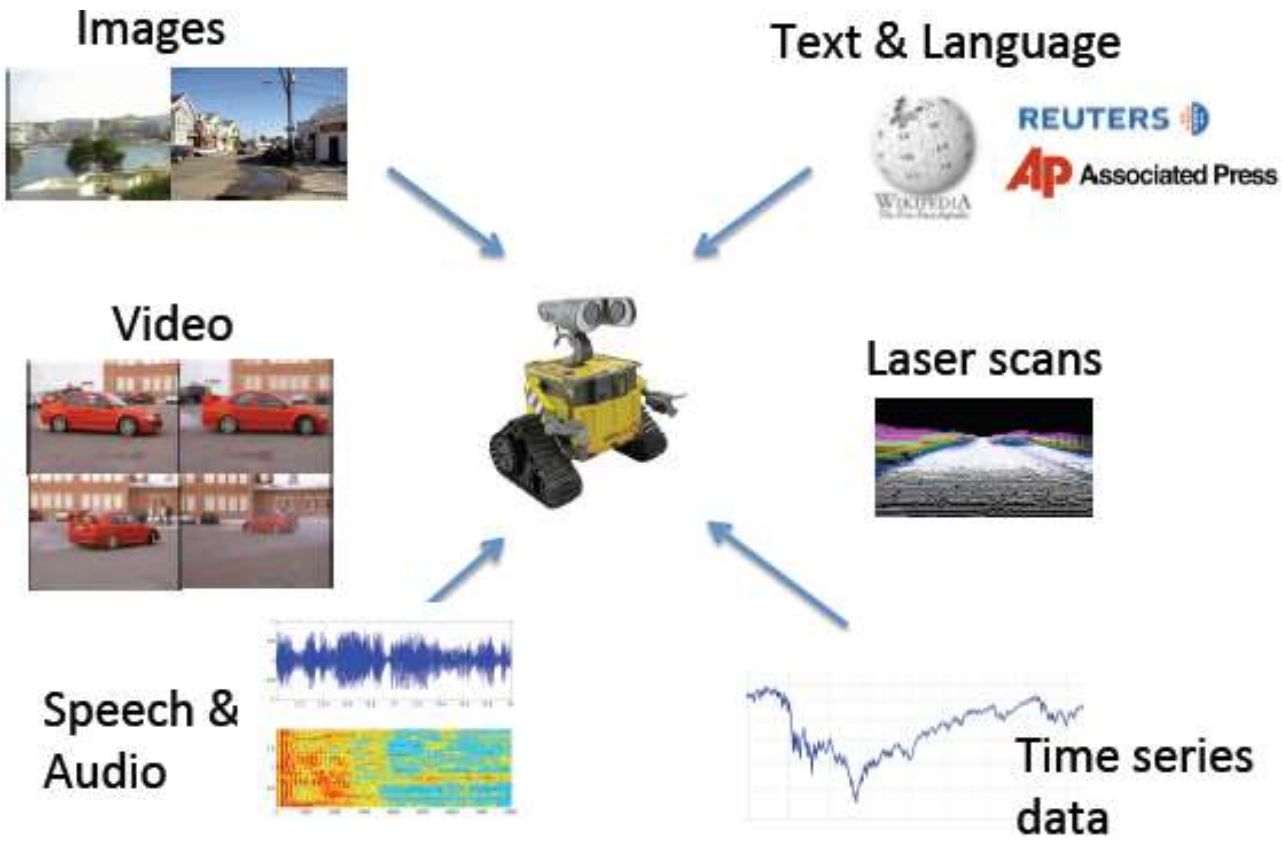
卷积神经网络CNN主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于CNN的特征检测层通过训练数据进行学习，所以在使用CNN时，避免了显式的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

流的分类方式几乎都是基于统计特征的，这就意味着在进行分辨前必须提取某些特征。然而，显式的特征提取并不容易，在一些应用问题中也并非总是可靠的。卷积神经网络避免了显式的特征取样，隐式地从训练数据中进行学习。这使得卷积神经网络明显有别于其他基于神经网络的分类器，通过结构重组和减少权值将特征提取功能融合进多层感知器。

卷积神经网络较一般神经网络在图像处理方面有如下优点： a) 输入图像和网络的拓扑结构能很好的吻合；b) 特征提取和模式分类同时进行，并同时在训练中产生；c) 权值共享能减少网络的训练参数，使神经网络结构变得更为简单，适应性更强。

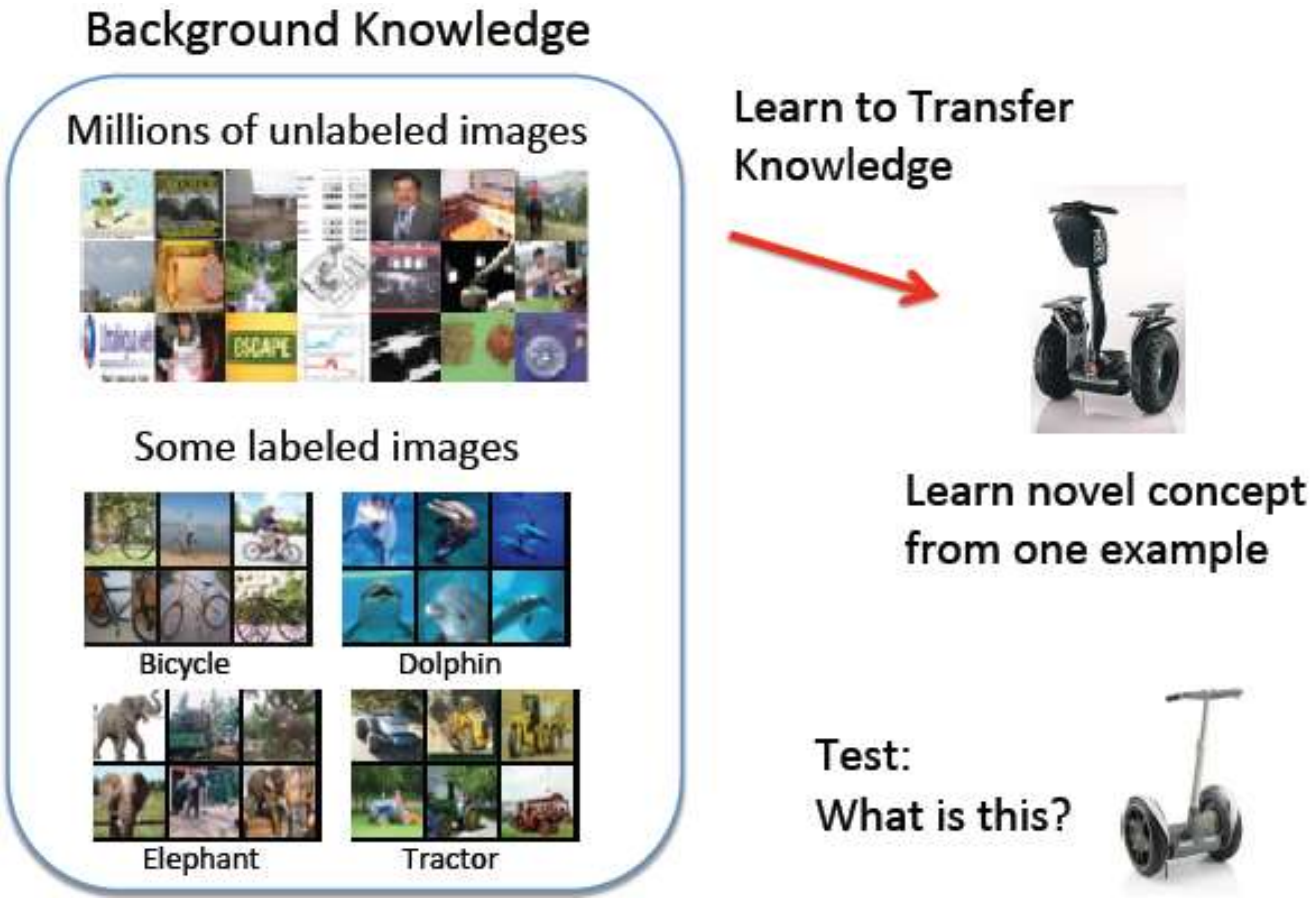
7.4 深度学习的应用

- 深度学习在多模态学习中的应用



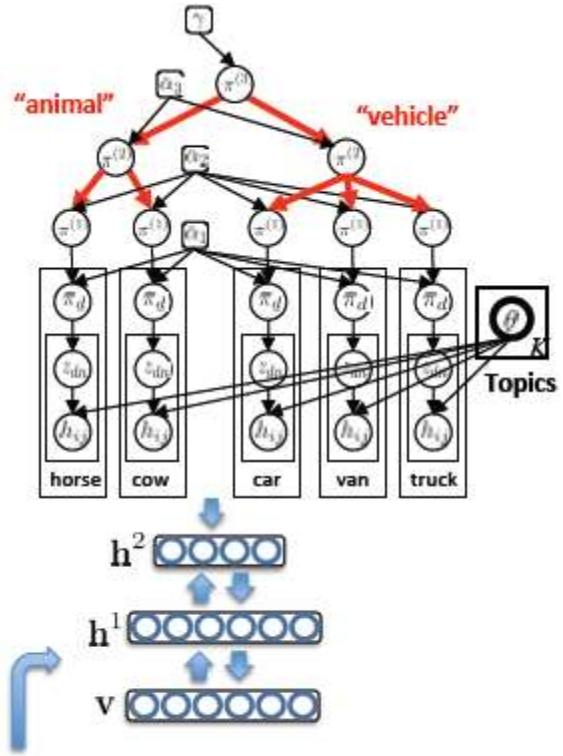
7.4 深度学习的应用

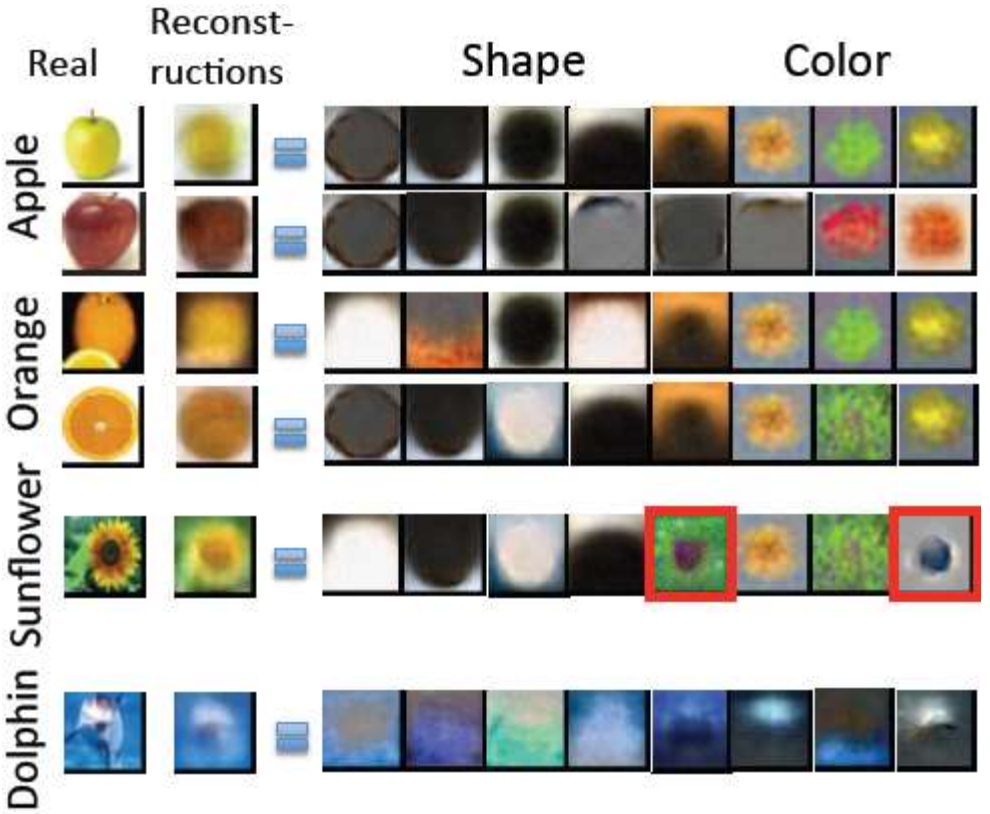
- 基于深度学习的迁移学习应用



7.4 深度学习的应用

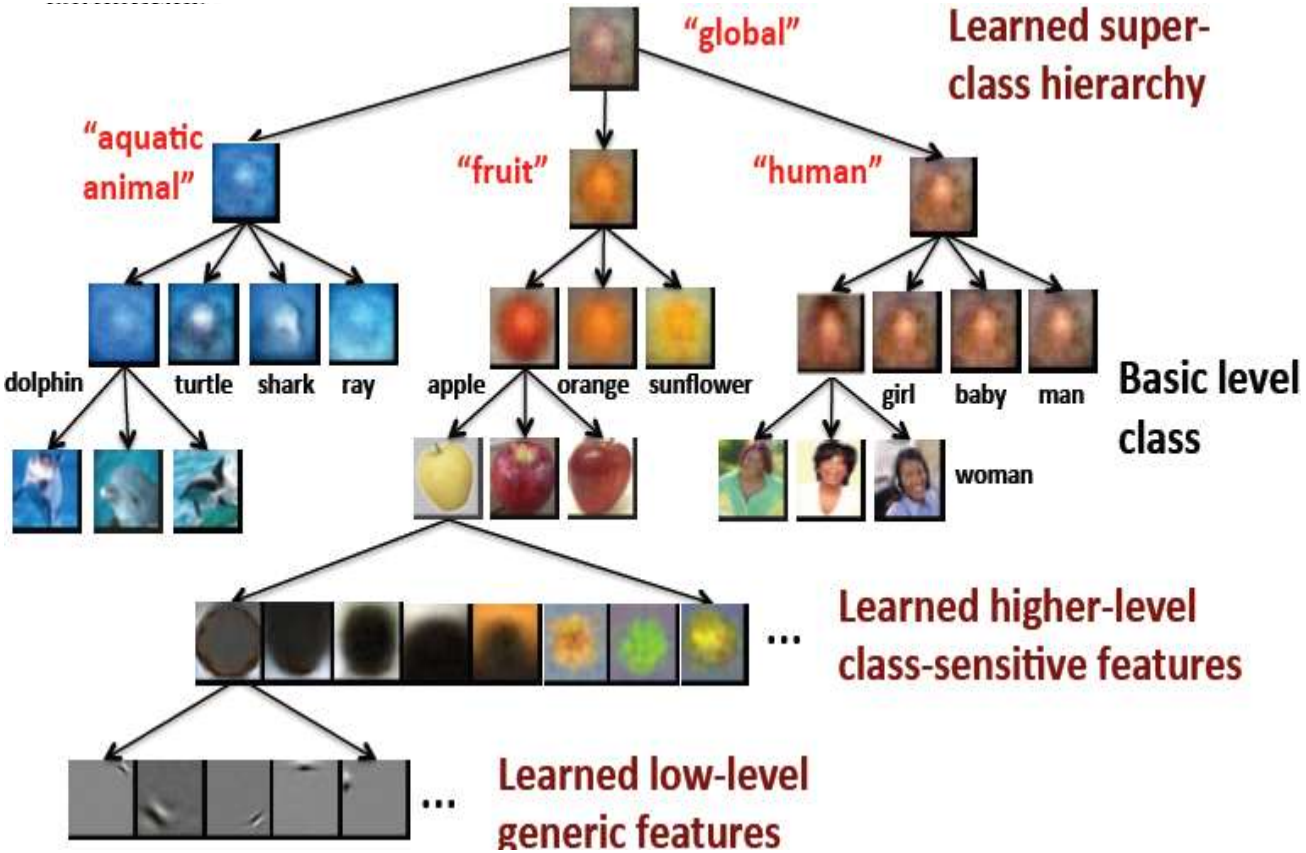
- 基于深度学习的迁移学习应用

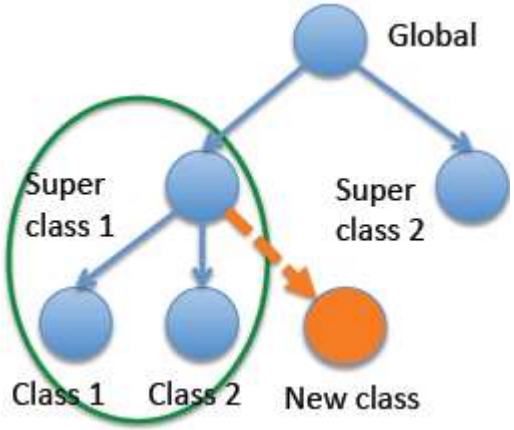




7.4 深度学习的应用

- 基于深度学习的迁移学习应用





7.4 深度学习的应用

- 深度学习在大尺度数据集上的应用

□大尺度数据集：

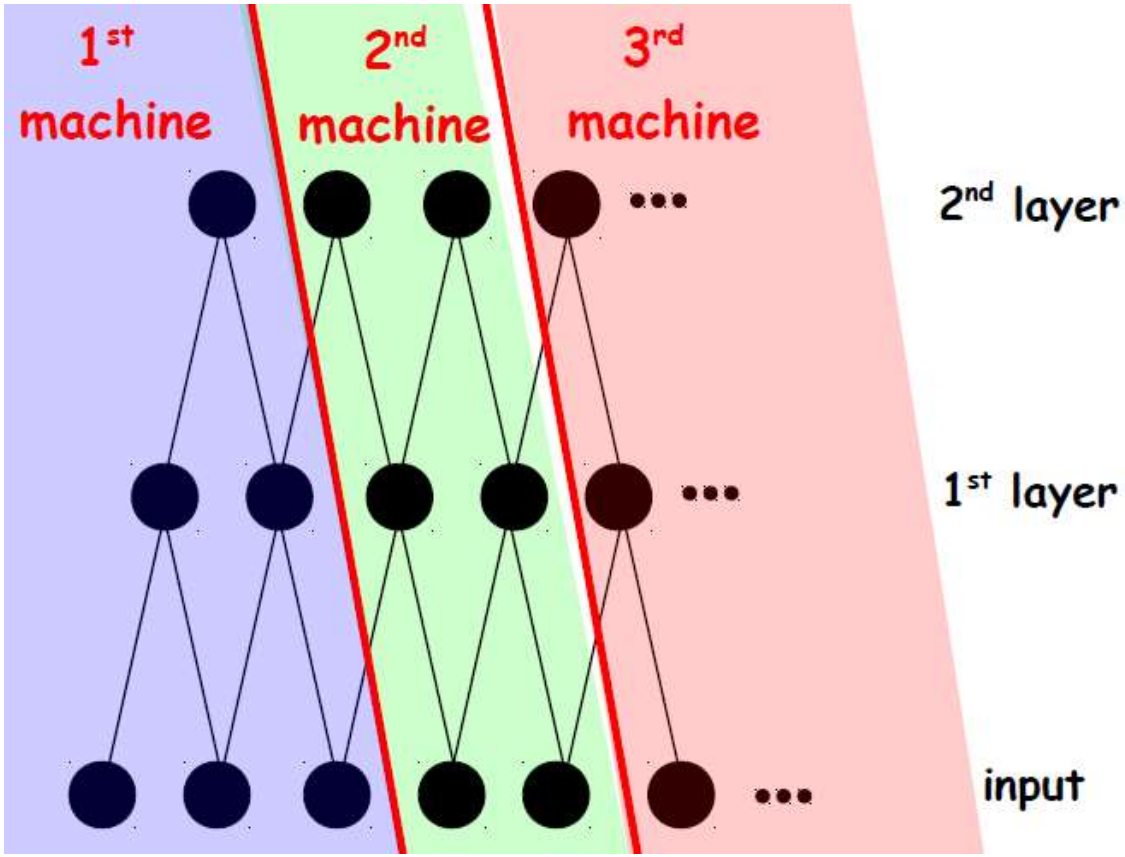
✓样本总数>100M；

✓类别总数>10K；

✓特征维度>10K。

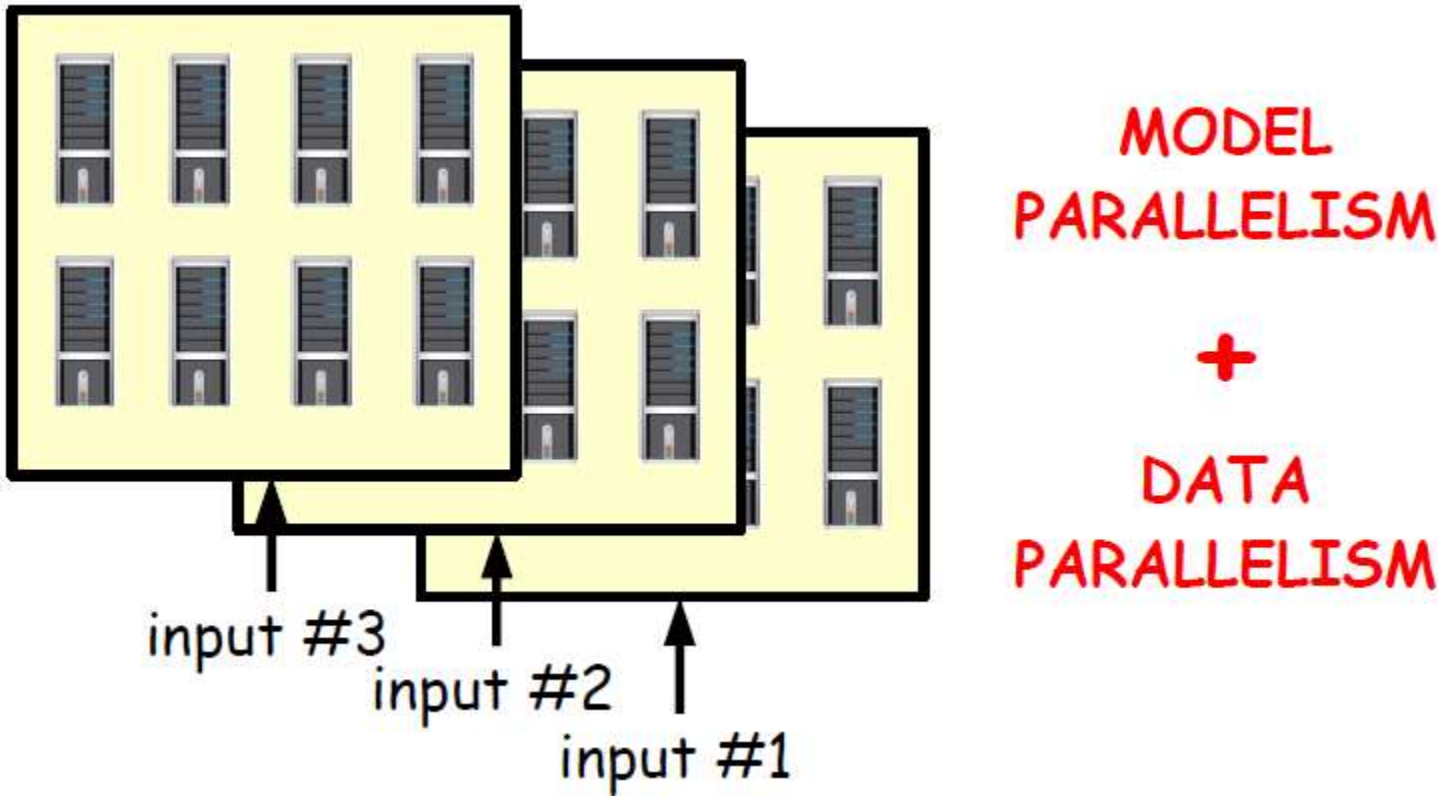
7.4 深度学习的应用

- 深度学习在大尺度数据集上的应用



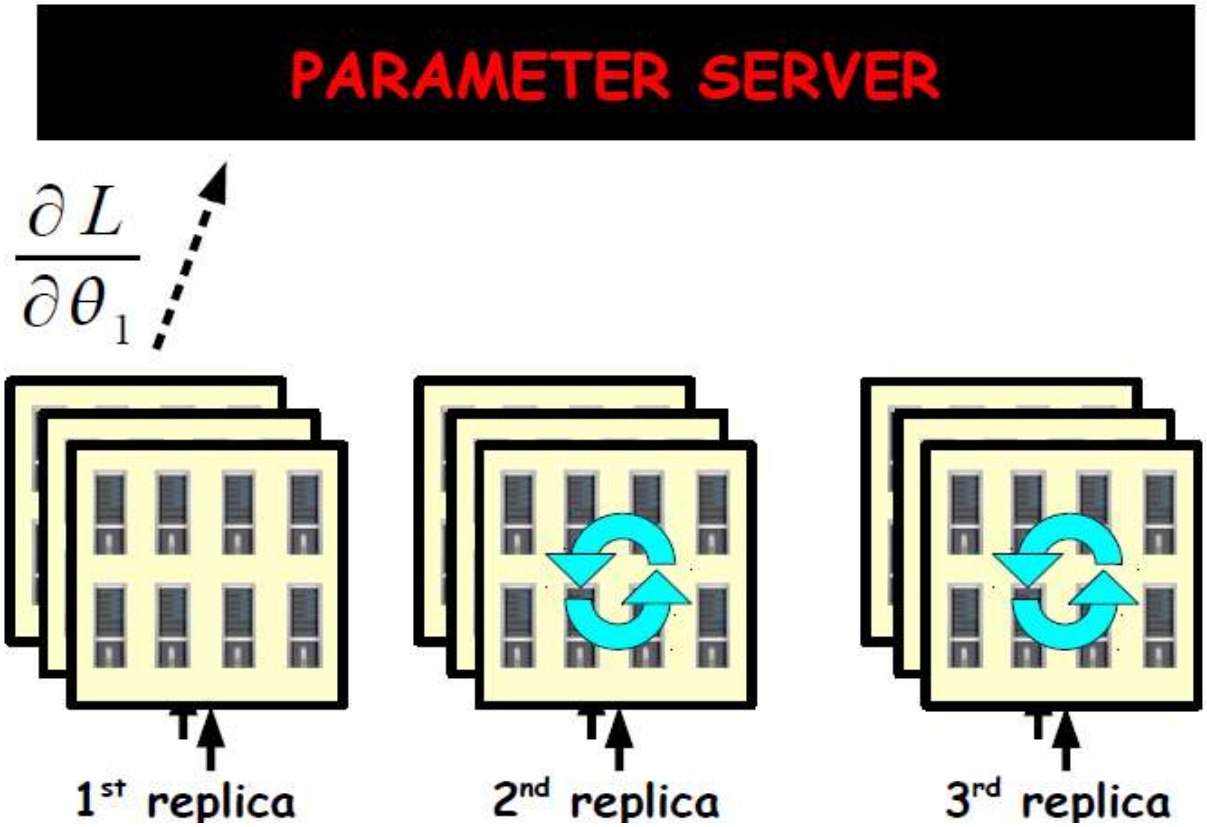
7.4 深度学习的应用

- 深度学习在大尺度数据集上的应用



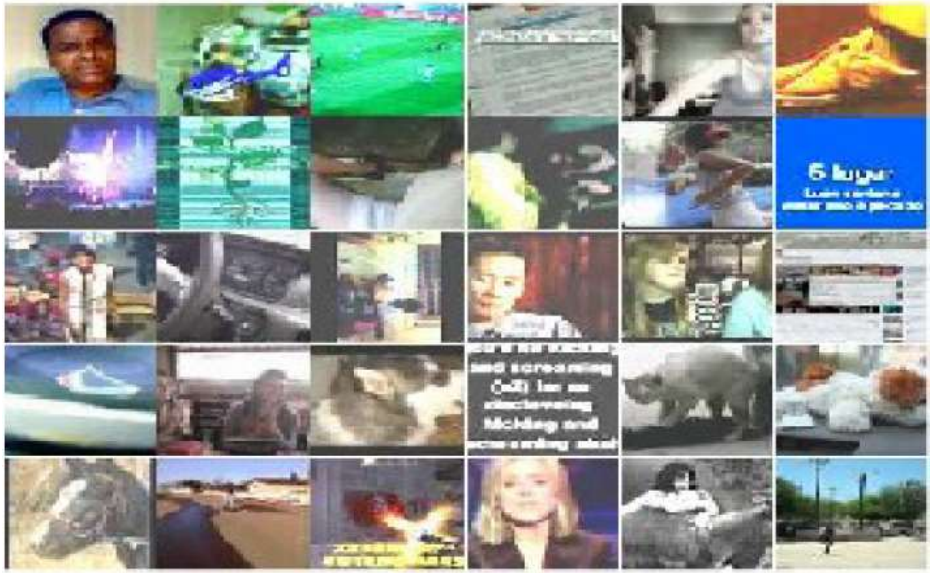
7.4 深度学习的应用

- 深度学习在大尺度数据集上的应用



7.4 深度学习的应用

- 深度学习在大尺度数据集上的应用



IMAGENET v.2011 (16M images, 20K categories)

METHOD	ACCURACY %
Weston & Bengio 2011	9.3
Linear Classifier on deep features	13.1
Deep Net (from random)	13.6
Deep Net (from unsup.)	15.8

参数个数达到1.15 billion，若不能并行优化参数，任务无法完成！

7.5 深度学习展望

未来需解决的问题：

- 对于一个特定的框架，多少维的输入它可以表现得较优？
- 对捕捉短时或者长时间的时间依赖，哪种架构才是有效的？
- 如何对于一个给定的深度学习架构，融合多种感知的信息？
- 如何分辨和利用学习获得的中、高层特征语义知识？
- 有什么正确的机理可以去增强一个给定的深度学习架构，以改进其鲁棒性和对变形及数据丢失的不变性？
- 模型方面是否有其他更为有效且有理论依据的深度模型学习算法？
- 是否存在更有效的可并行训练算法？

7.5 深度学习展望

课外思考题：

1、以ORL人脸数据库为例，给出采用BP神经网络进行人脸识别的方法步骤。

2、以ORL人脸数据库为例，给出采用CNN网络进行人脸识别的方法步骤。

3、以手写体数字识别为例，给出采用CNN网络进行数字识别的方法步骤。

4、以ORL人脸数据库为例，试编写用PCA对其进行特征提取并用BP神经网络进行识别的MATLAB程序(*: 选做)。

End of This Chapter.