模式识别与机器学习

Pattern Recognition and Machine Learning

课程内容

- **模式识别与机器学习概述**
- **模式识别与机器学习的基本方法**
  - 回归分析、线性判别函数、神经网络、核方法和支持向量机、决策树分类、逻辑斯特回归
  - 贝叶斯统计决策理论、概率密度函数估计
  - 无监督学习和聚类
  - 特征选择与提取

神经网络发展史

第一阶段

- 1943年, McCulloch和Pitts 提出第一个神经元数学模型, 即M-P模型, 并从原理上证明了人工神经网络能够计算任何算数和逻辑函数
- 1949年, Hebb 发表《The Organization of Behavior》一书, 提出生物神经元学习的机理, 即Hebb学习规则
- 1958年, Rosenblatt 提出感知机网络（Perceptron）模型和其学习规则
- 1960年, Widrow和Hoff提出自适应线性神经元（Adaline）模型和最小均方学习算法
- 1969年, Minsky和Papert 发表《Perceptrons》一书, 指出单层神经网路不能解决非线性问题, 多层网络的训练算法尚无希望. 这个论断导致神经网络进入低谷

神经网络发展史

第二阶段

1982年, 物理学家Hopfield提出了一种具有联想记忆、优化计算能力的递归网络模型, 即Hopfield 网络

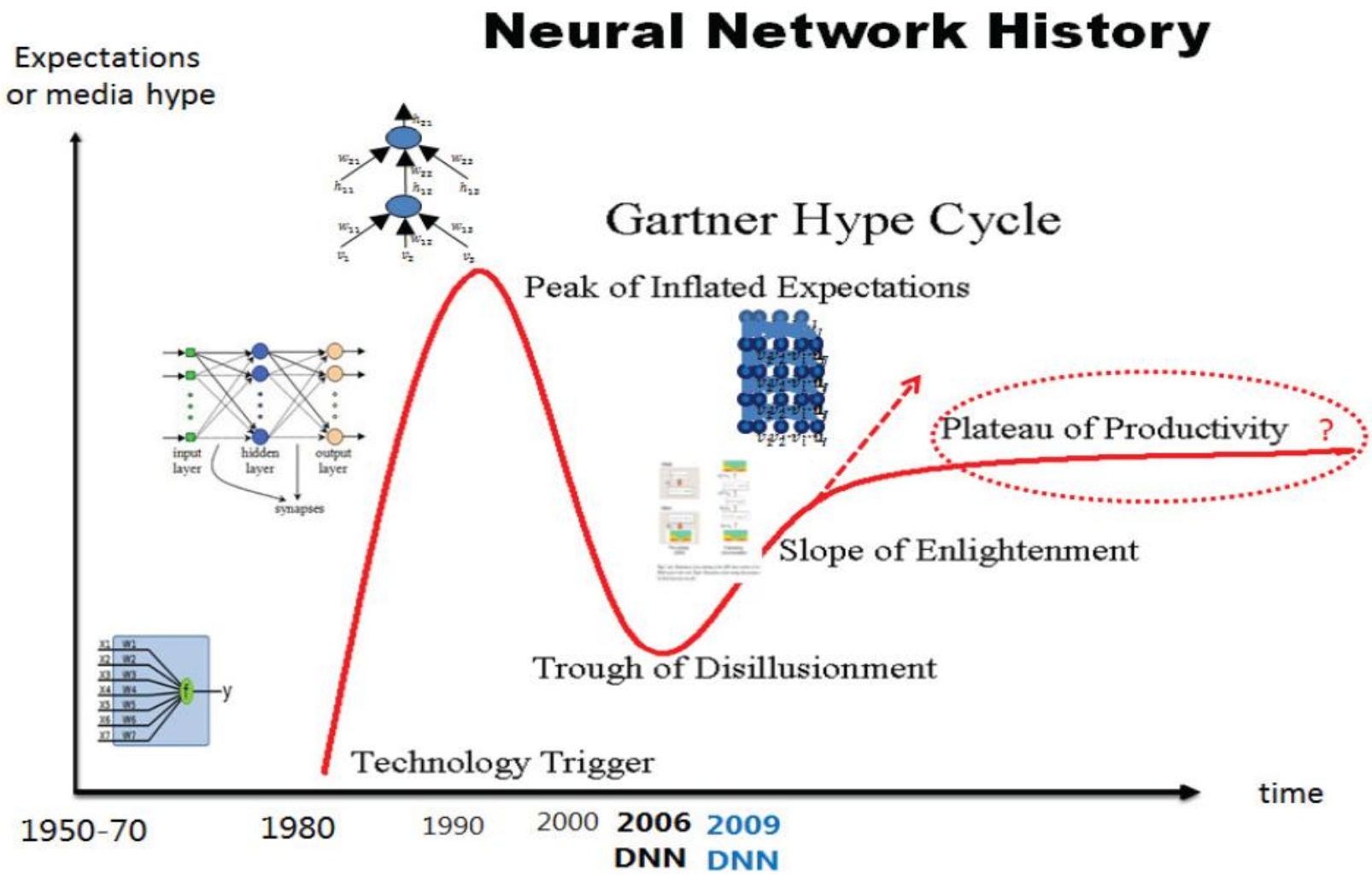
- 1986年, Rumelhart 等编辑的著作《Parallel Distributed Proceesing:Explorations in the Microstructures of Cognition》报告了反向传播算法
- 1987年, IEEE 在美国加州圣地亚哥召开第一届神经网络国际会议（ICNN）
- 90年代初, 伴随统计学习理论和SVM的兴起, 神经网络由于理论不够清楚, 试错性强, 难以训练, 再次进入低谷

神经网络发展史

第三阶段

- 2006年, Hinton提出了深度信念网络(DBN), 通过“预训练+微调”使得深度模型的最优化变得相对容易
- 2012年, Hinton 组参加ImageNet 竞赛, 使用 CNN 模型以超过第二名10个百分点的成绩夺得当年竞赛的冠军
- 伴随云计算、大数据时代的到来，计算能力的大幅提升，使得深度学习模型在计算机视觉、自然语言处理、语音识别等众多领域都取得了较大的成功

神经网络发展史


Neural Network History
Expectations
or media hype
Gartner Hype Cycle
Peak of Inflated Expectations
Plateau of Productivity ?
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
input layer
hidden layer
output layer
synapses
time
1950-70
1980
1990
2000
2006 DNN
2009 DNN

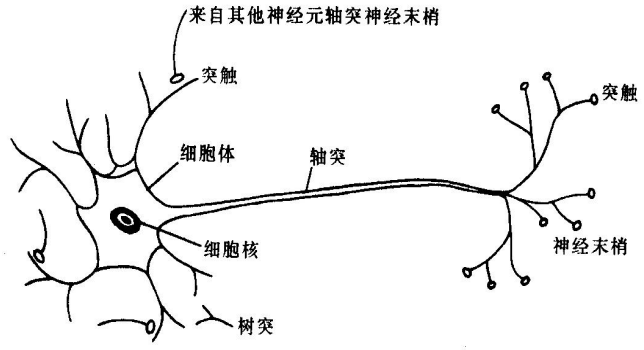
神经元模型

■ 神经网络的定义

“神经网络是由具有适应性的简单单元组成的广泛并行互联的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的反应”

[Kohonen, 1988]

- 机器学习中的神经网络通常是指“神经网络学习”或者机器学习与神经网络两个学科的交叉部分
- 神经元模型即上述定义中的“简单单元”是神经网络的基本成分
- 生物神经网络：每个神经元与其他神经元相连，当它“兴奋”时，就会向相连的神经云发送化学物质，从而改变这些神经元内的电位；如果某神经元的电位超过一个“阈值”，那么它就会被激活，即“兴奋”起来，向其它神经元发送化学物质



神经元模型

■ M-P 神经元模型 [McCulloch and Pitts，1943]

- 输入：来自其他 个神经元传递过来的输入信号
- 处理：输入信号通过带权重的连接进行传递，神经元接受到总输入值将与神经元的阈值进行比较
- 输出：通过激活函数的处理以得到输出

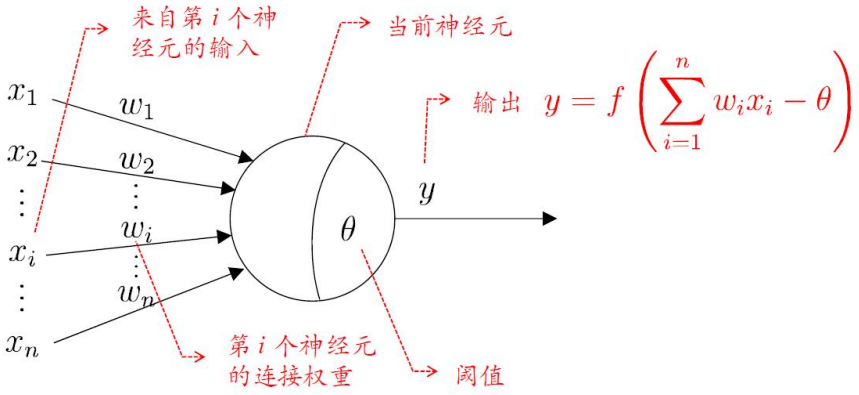
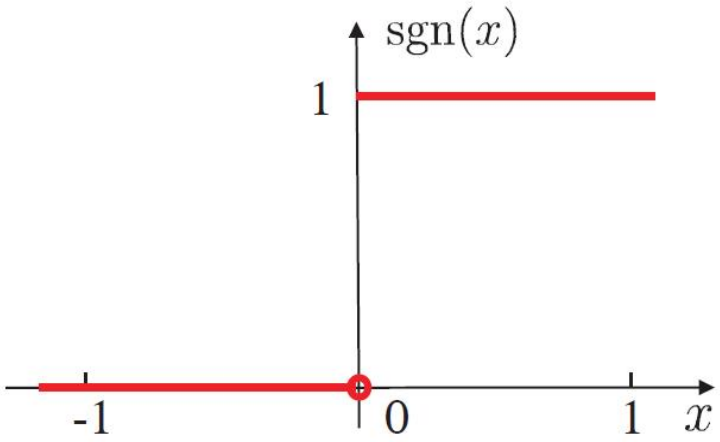


图 5.1 M-P 神经元模型

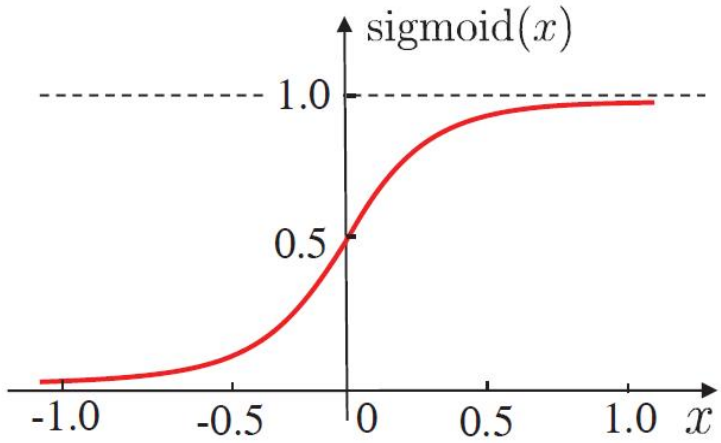
神经元模型

■ 激活函数



$$\mathrm{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases}$$

(a) 阶跃函数



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

(b) Sigmoid 函数

典型的神经元激活函数

感知机与多层网络

■ 感知机

- 感知机由两层神经元组成，输入层接受外界输入信号传递给输出层，输出层是M-P神经元（阈值逻辑单元）
- 感知机能够容易地实现逻辑与、或、非运算

- “与” $x_1 \wedge x_2$：令 $w_1 = w_2 = 1, \theta = 0$，则 $y = f(1 \cdot x_1 + 1 \cdot x_2 - 2)$；仅在 $x_1 = x_2 = 1$ 时 $y = 1$
- “或” $x_1 \vee x_2$：令 $w_1 = w_2 = 1, \theta = 0.5$ 则 $y = f(1 \cdot x_1 + 1 \cdot x_2 - 2)$ 仅在 $x_1 = 1$ 或者 $x_2 = 1$ 时 $y = 1$；
- “非” $\neg x_1$ 令 $w_1 = -0.6, w_2 = 0, \theta = -0.5$ 则当 $x_1 = 1$ 时，$y = 0$；当 $x_1 = 0$，$y = 1$

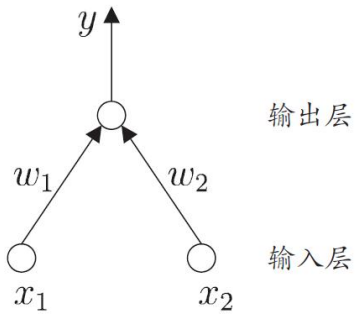


图 5.3 两个输入神经元的感知机网络结构示意图

感知机与多层网络

■ 感知机

- 若两类模式线性可分, 则感知机的学习过程一定会收敛；否感知机的学习过程将会发生震荡

  [Minsky and Papert, 1969]

- 单层感知机的学习能力非常有限, 只能解决线性可分问题

- 事实上, 与、或、非问题是线性可分的, 因此感知机学习过程能够求得适当的权值向量. 而异或问题不是线性可分的, 感知机学习不能求得合适解

- 对于非线性可分问题, 如何求解?

  多层感知机

感知机与多层网络

■ 多层感知机

□ 解决异或问题的两层感知机

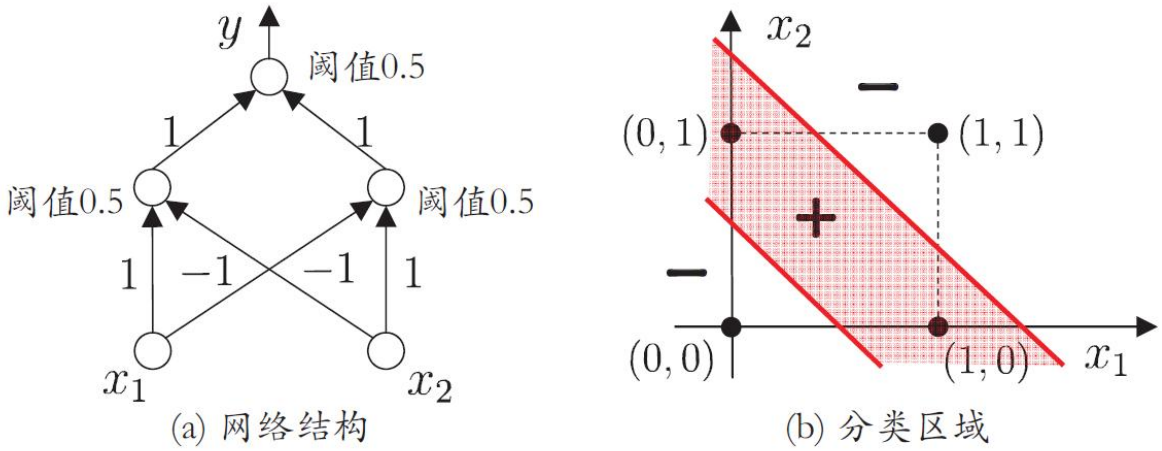


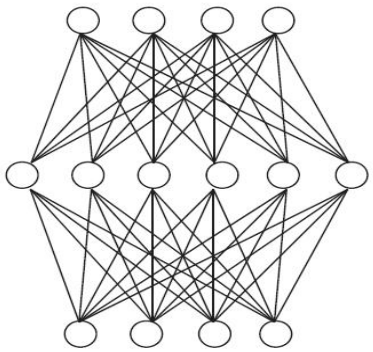
图 5.5 能解决异或问题的两层感知机

- 输出层与输入层之间的一层神经元, 被称之为隐层或隐含层, 隐含层和输出层神经元都是具有激活函数的功能神经元

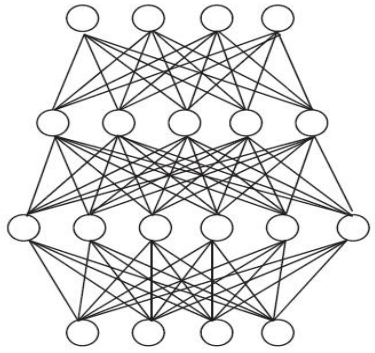
感知机与多层网络

■ 多层前馈神经网络

- 定义：每层神经元与下一层神经元全互联，神经元之间不存在同层连接也不存在跨层连接
- 前馈：输入层接受外界输入，隐含层与输出层神经元对信号进行加工，最终结果由输出层神经元输出
- 学习：根据训练数据来调整神经元之间的“连接权”以及每个功能神经元的“阈值”
- 多层网络：包含隐层的网络



(a) 单隐层前馈网络



(b) 双隐层前馈网络

感知机与多层网络

■ 误差逆传播算法

误差逆传播算法（Error BackPropagation, 简称BP）是最成功的训练多层前馈神经网络的学习算法.

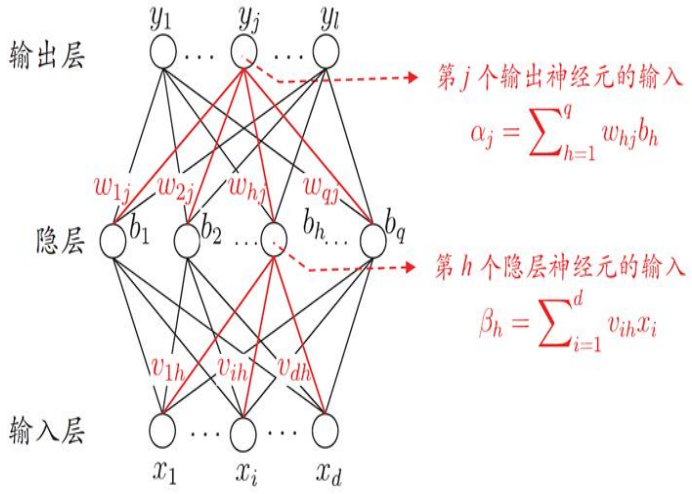
- 给定训练集 $D=\{(\boldsymbol{x}_i, y_i)\}, \boldsymbol{x}_i \in R^d, y_i \in R^l, (i=1,2,\cdots,m)$ 即输入示例由d个属性描述, 输出 l 维实值向量.
- 为方便讨论, 给定一个拥有 d个输入神经元, l 个输出神经元, q 个隐层神经元的多层前向前馈网络结构.
- 记号:

θ_j： 输出层第 j 个神经元阈值;

γ_h： 隐含层第 h 个神经元阈值;

v_{ih}： 输入层与隐层神经元之间的连接权重;

w_{hj}： 隐层与输出层神经元之间的连接权重;



感知机与多层网络

■ 误差逆传播算法（BP算法）

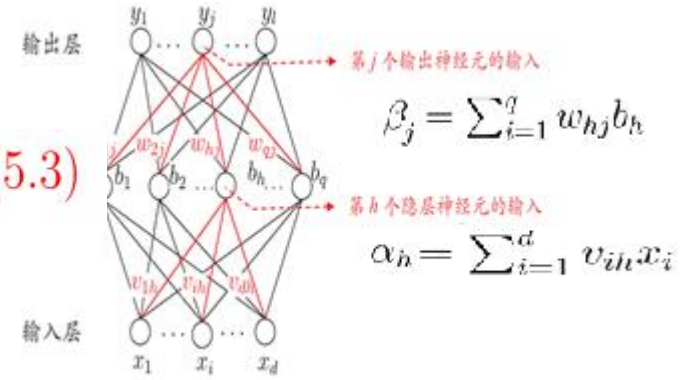
对于样例 (x_k, y_k)**,假设网络的实际输出为** $\hat{y}_k$

前向计算:

step1: $b_h = f(\alpha_h - \gamma_h),\ \alpha_h = \sum_{i=1}^{d} v_{ih} x_i$

step2: $\hat{y}_j^k = f(\beta_j - \theta_j),\ \beta_j = \sum_{i=1}^{q} w_{hj} b_h$ (5.3)

step3: $E_k = \frac{1}{2} \sum_{j=1}^{l} (\hat{y}_j^k - y_j^k)^2$



参数数目:

权重: v_{ih} ,w_{hj} 阈值:θ_j ,γ_h $(i = 1, \cdots, d, h = 1, , \cdots, q, j = 1, \cdots, l)$

因此网络中需要$(d + l + 1)q + l$ 个参数需要优化

参数优化:

BP是一个迭代学习算法, 在迭代的每一轮中采用广义的感知机学习对参数进行更新估计,任意的参数 v 的更新估计式为

$$v \leftarrow v + \triangle v.$$

感知机与多层网络

■ 误差逆传播算法（BP算法）

- BP算法基于梯度下降策略, 以目标的负梯度方向对参数进行调整. 对误差 E_k ，给定学习率 η

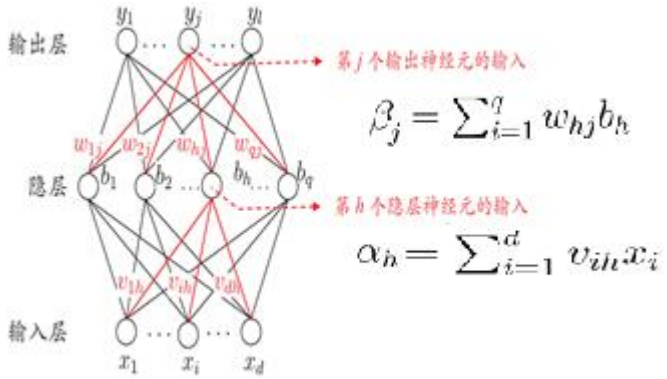
$$\triangle w_{hj} = -\eta \frac{\partial E_k}{\partial w_{jk}}$$

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

$$g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}$$

$$= -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j)$$

$$= \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \quad (5.10)$$



感知机与多层网络

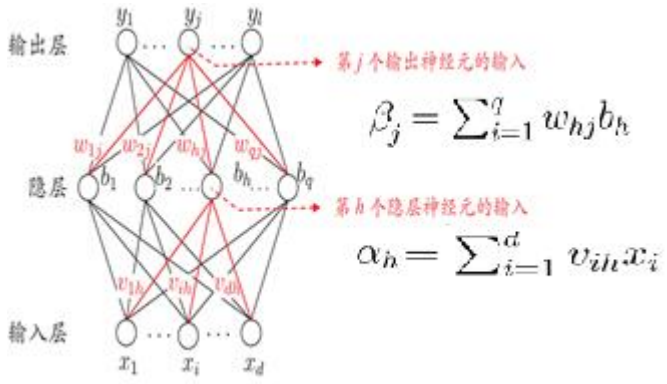
■ 误差逆传播算法（BP算法）

类似的可以推导出：

$$\triangle\theta_j = -\eta g_j bh \quad (5.12)$$

$$\triangle v_{ih} = \eta e_h x_i \quad (5.13)$$

$$\triangle\gamma_h = -\eta e_h \quad (5.14)$$



其中

$$e_h = -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} = \sum_{j=1}^{h} w_{hj} g_j f'(\alpha_h - \gamma_h)$$

$$= -\sum_{j=1}^{l} \frac{\partial E_k}{\partial \beta_j} \partial b_h f'(\alpha_h - \gamma_h) = b_h(1-b_h)\sum_{j=1}^{h} w_{hj} g_j. \quad (5.15)$$

- 学习率 $\eta \in (0,1)$ 控制着算法每一轮迭代中的更新步长, 若太长则让容易震荡, 太小则收敛速度又会过慢.

感知机与多层网络

■ 误差逆传播算法（BP算法）

输入： 训练集 $D = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{m}$；
学习率 η.

过程：

1: 在$(0,1)$范围内随机初始化网络中所有连接权和阈值
2: **repeat**
3: **for all** $(\boldsymbol{x}_k, \boldsymbol{y}_k) \in D$ **do**
4: 根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\boldsymbol{y}}_k$;
5: 根据式(5.10) 计算输出层神经元的梯度项 g_j;
6: 根据式(5.15) 计算隐层神经元的梯度项 e_h;
7: 根据式(5.11)-(5.14) 更新连接权 w_{hj}, v_{ih} 与阈值 θ_j, γ_h
8: **end for**
9: **until** 达到停止条件

输出： 连接权与阈值确定的多层前馈神经网络

误差逆转波算法

感知机与多层网络

■ 误差逆传播算法（BP算法）

□标准 BP 算法

- 每次针对单个训练样例更新权值与阈值.
- 参数更新频繁,不同样例可能抵消, 需要多次迭代.

□累计 BP 算法

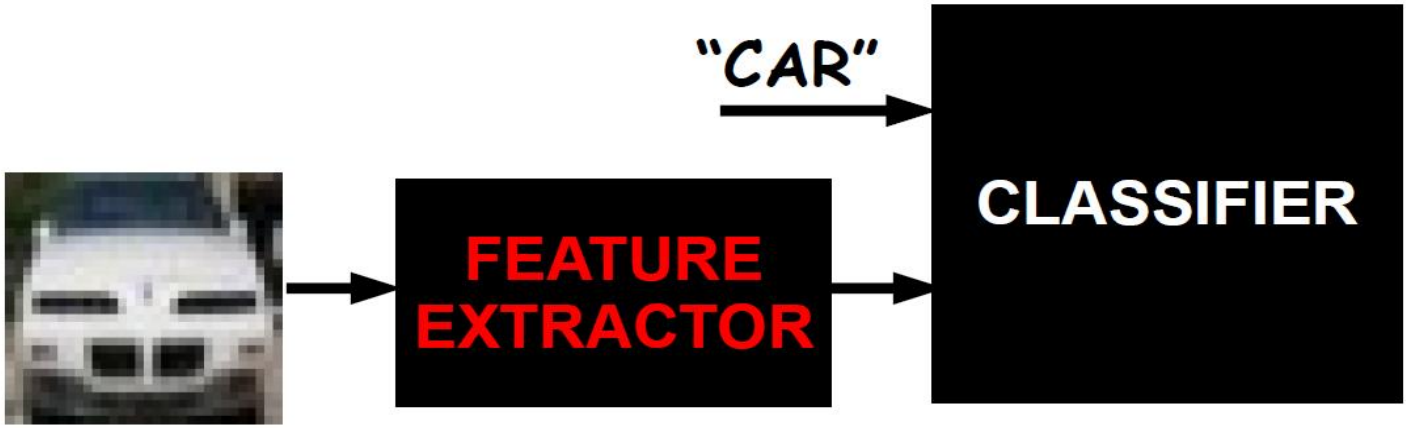
- 其优化的目标是最小化整个训练集上的累计误差
- 读取整个训练集一遍才对参数进行更新, 参数更新频率较低.

□实际应用

但在很多任务中, 累计误差下降到一定程度后, 进一步下降会非常缓慢, 这时标准BP算法往往会获得较好的解, 尤其当训练集非常大时效果更明显.

感知机与多层网络

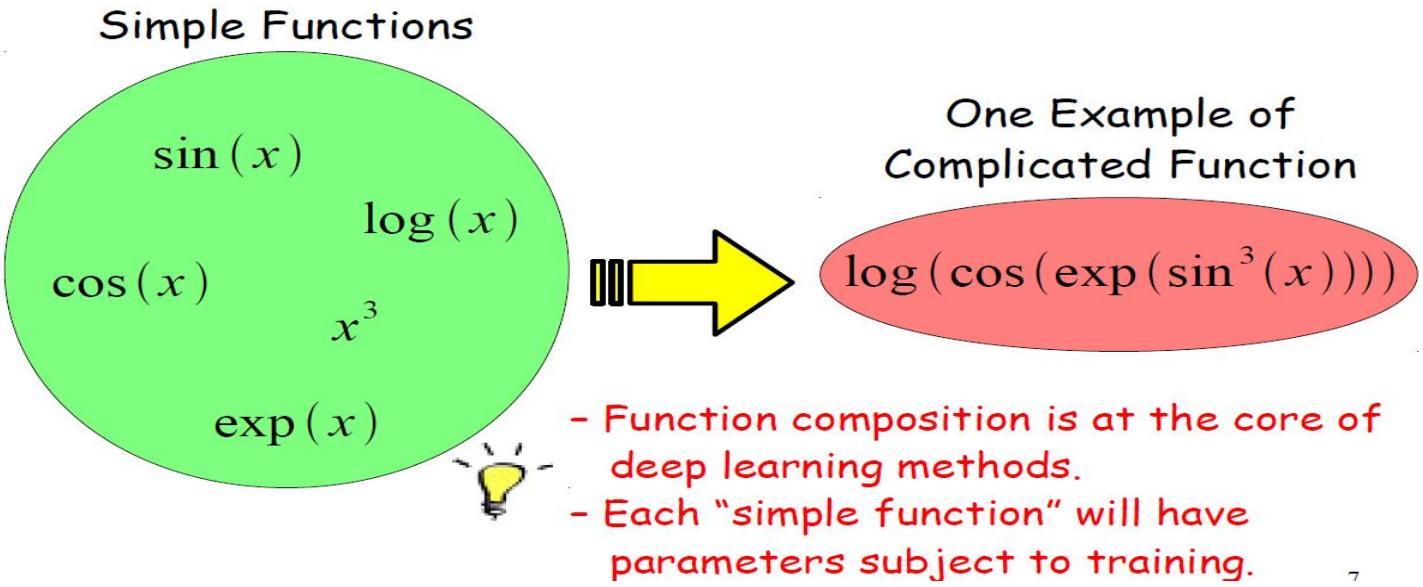
- **误差逆传播算法（BP算法）**
  - 多层前馈网络表示能力



IDEA: Use data to optimize features for the given task.

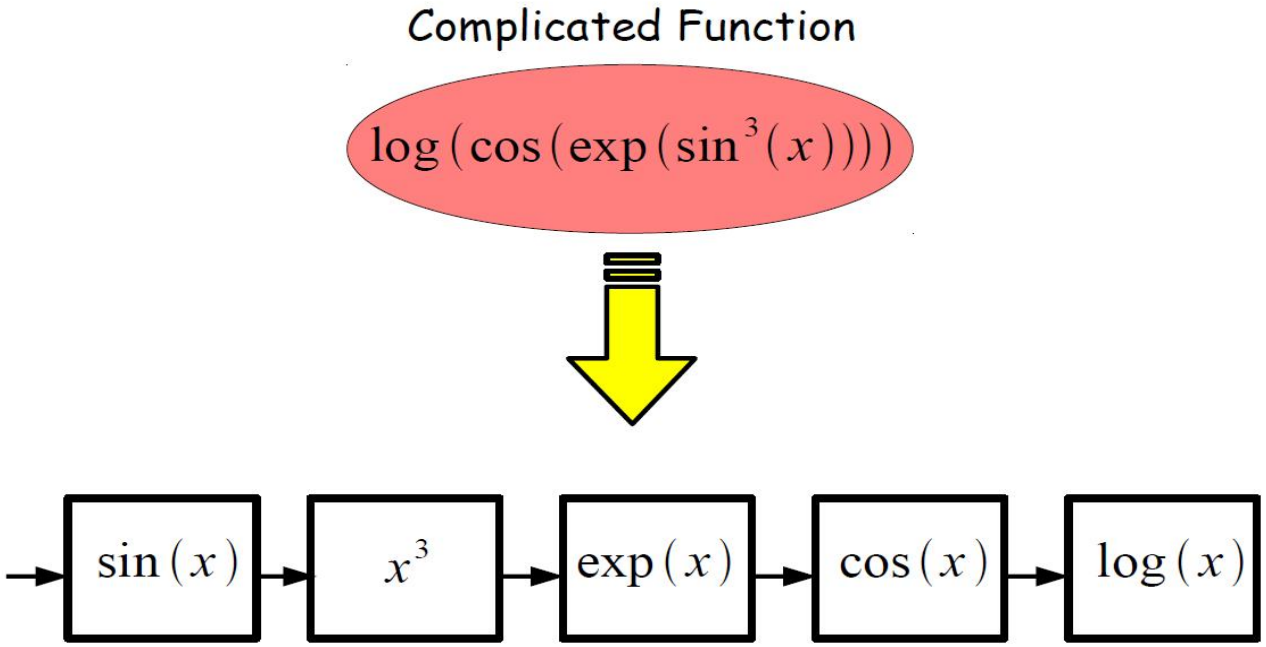
感知机与多层网络

- **误差逆传播算法（BP算法）**
  - 多层前馈网络表示能力



感知机与多层网络

- **误差逆传播算法（BP算法）**
  - 多层前馈网络表示能力



感知机与多层网络

■ 误差逆传播算法（BP算法）

□多层前馈网络表示能力

只需要一个包含足够多神经元的隐层, 多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 **[Hornik et al.，1989]**

多层前馈网络局限

- 神经网络由于强大的表示能力, 经常遭遇过拟合. 表现为：训练误差持续降低, 但测试误差却可能上升
- 如何设置隐层神经元的个数仍然是个未决问题. 实际应用中通常使用“试错法”调整

□缓解过拟合的策略

- 早停：在训练过程中, 若训练误差降低, 但验证误差升高, 则停止训练
- 正则化：在误差目标函数中增加一项描述网络复杂程度的部分, 例如连接权值与阈值的平方和

感知机与多层网络

RBF 网络 [Broomhead and Lowe, 1988]

- RBF 网络是一种单隐层前馈神经网络, 它使用径向基函数作为隐层神经元激活函数, 而输出层则是隐层神经元输出的线性组合.
- RBF网络模型

假定输入为 d 维的向量 $\boldsymbol{x}$, 输出为实值, 则RBF 网络可以表示为

$$\varphi(\boldsymbol{x}) = \sum_{i=1}^{q} w_i \rho(\boldsymbol{x}, \boldsymbol{c}_i)$$

其中 q 为隐层神经元的个数,$\boldsymbol{c}_i$ 和w_i 分别是第 i 神经元对应的中心和权重 $\rho(\boldsymbol{x}, \boldsymbol{c}_i)$ 是径向基函数.
常用的高斯径向基函数形如

$$\rho(\boldsymbol{x}, \boldsymbol{c}_i) = e^{-\beta_i \|\boldsymbol{x}-\boldsymbol{c}_i\|^2}$$

感知机与多层网络

RBF 网络 [Broomhead and Lowe, 1988]

- RBF网络性质

具有足够多隐层神经元RBF 神经网络能以任意精度逼近任意连续函数.

[Park and Sandberg，1991]

- RBF网络训练
  - Step1:确定神经元中心，常用的方式包括随机采样、聚类等
  - Step2:利用BP算法等确定参数

感知机与多层网络

ATR 网络

◘ 竞争学习

竞争学习是神经网络中一种常用的无监督学习策略, 在使用该策略时, 网络的输出神经元相互竞争, 每一时刻仅有一个神经元被激活, 其他神经元的状态被抑制.

◘ ART 网络 [Carpenter and Grossberg, 1987]

- ART 网络是竞争学习的重要代表
- ART 网络由比较层、识别层、识别阈值和重置模块构成
- 比较层负责接收输入样本, 并将其传送给识别层神经元
- 识别层每个神经元对应一个模式类, 神经元的数目可在训练过程中动态增长以增加新的模式类

深度学习

深度学习模型

- 典型的深度学习模型就是很深层的神经网络.

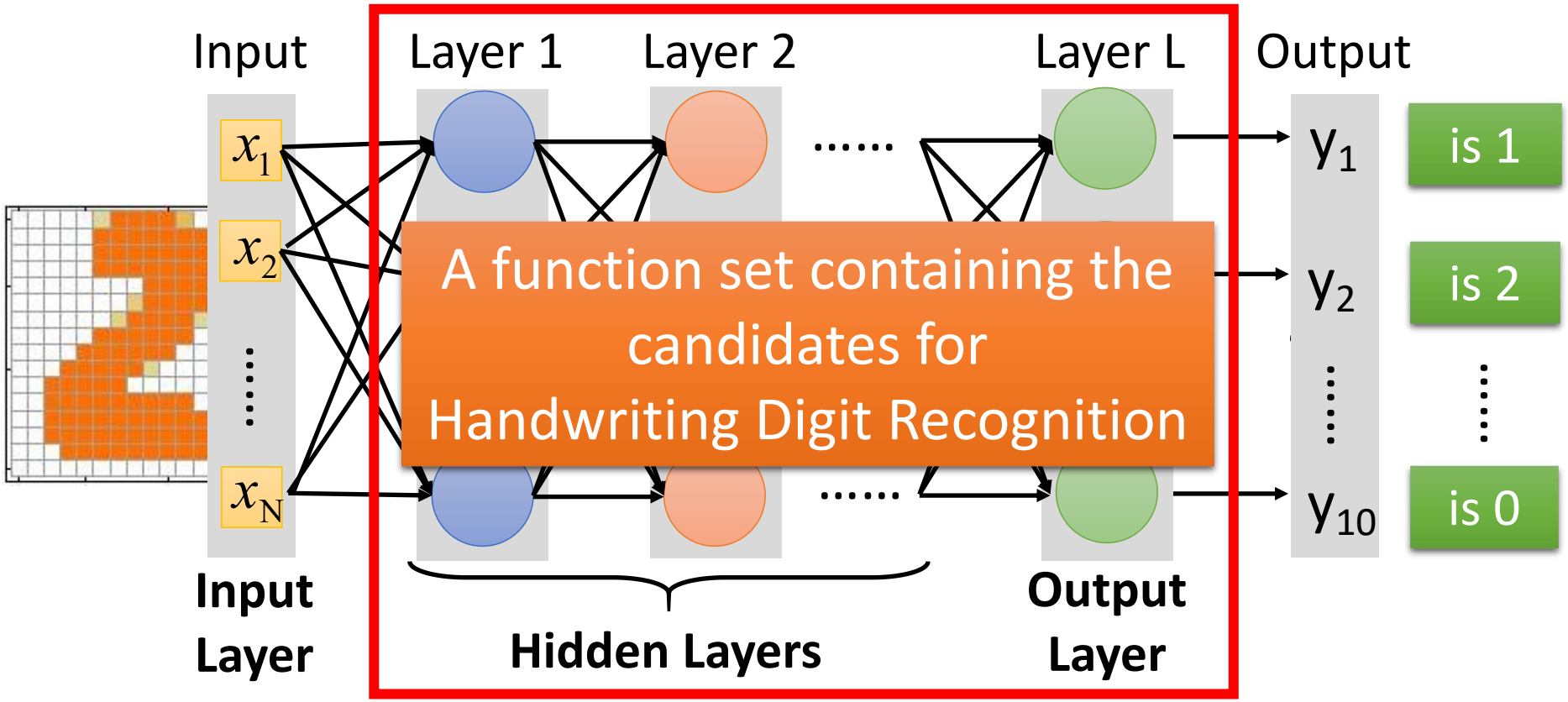
□ 模型复杂度

- 增加隐层神经元的数目 (模型宽度)
- 增加隐层数目 （模型深度）
- 从增加模型复杂度的角度看, 增加隐层的数目比增加隐层神经元的数目更有效. 这是因为增加隐层数不仅增加额拥有激活函数的神经元数目, 还增加了激活函数嵌套的层数.

□ 复杂模型难点

多隐层网络难以直接用经典算法（例如标准BP算法）进行训练, 因为误差在多隐层内逆传播时, 往往会”发散”而不能收敛到稳定状态.

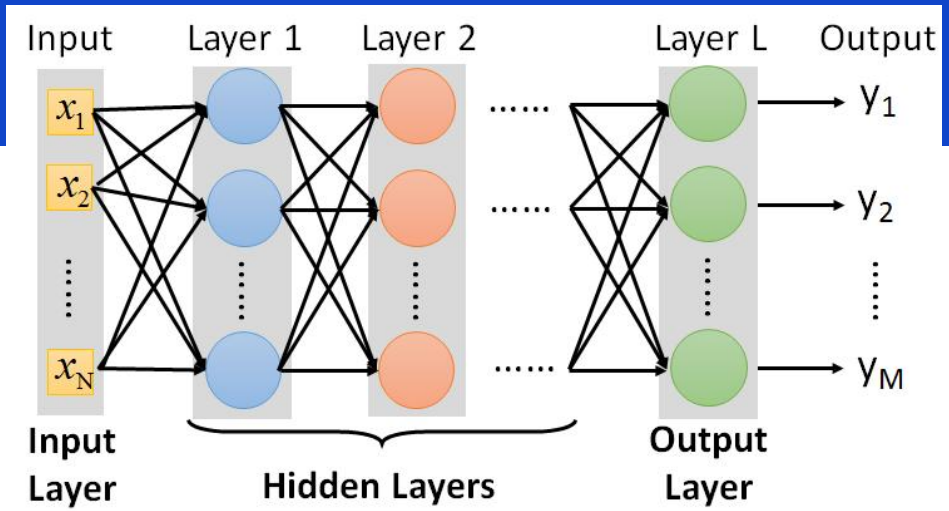
Example Application



You need to decide the network structure to let a good function in your function set.

参考自李宏毅老师课件

FAQ



- Q: How many layers? How many neurons for each layer?

Trial and Error + Intuition

- Q: Can the structure be automatically determined?
  - E.g. Evolutionary Artificial Neural Networks
- Q: Can we design the network structure?

Convolutional Neural Network (CNN)

参考自李宏毅老师课件

复杂模型训练方法

- 权共享
- 一组神经元使用相同的连接权值.
- 权共享策略在卷积神经网络(CNN)[LeCun and Bengio, 1995; LeCun et al., 1998]中发挥了重要作用.

- 卷积神经网络

结构：CNN复合多个卷积层和采样层对输入信号进行加工, 然后在连接层实现与输出目标之间的映射.

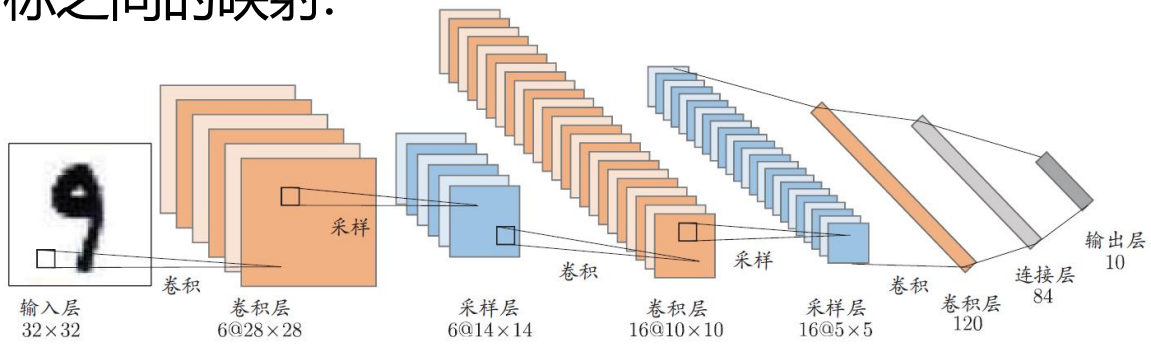


图 5.15 卷积神经网络用于手写数字识别 [LeCun et al., 1998]

深度学习

卷积神经网络

- 卷积层：每个卷积层包含多个特征映射，每个特征映射是一个由多个神经元构成的“平面”，通过一种卷积滤波器提取的一种特征
- 采样层：亦称“汇合层”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息
- 连接层：每个神经元被全连接到上一层每个神经元，本质就是传统的神经网络，其目的是通过连接层和输出层的连接完成识别任务

卷积神经网络激活函数

在CNN中通常将 sigmoid 激活函数替换为修正的线性函数

卷积神经网络训练

CNN 可以用BP进行训练，但在训练中，无论是卷积层还是采样层，每一组神经元都是用相同的连接权，从而大幅减少了需要训练的参数数目

深度学习

卷积神经网络

- Some patterns are much smaller than the whole image

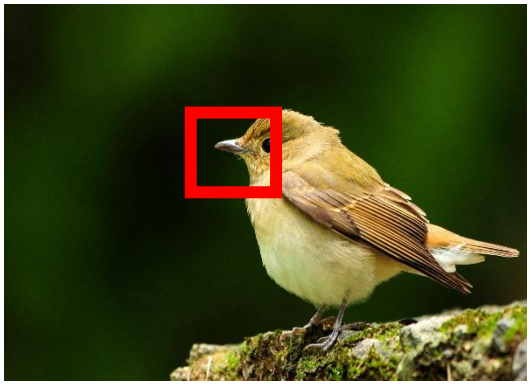
A neuron does not have to see the whole image to discover the pattern.

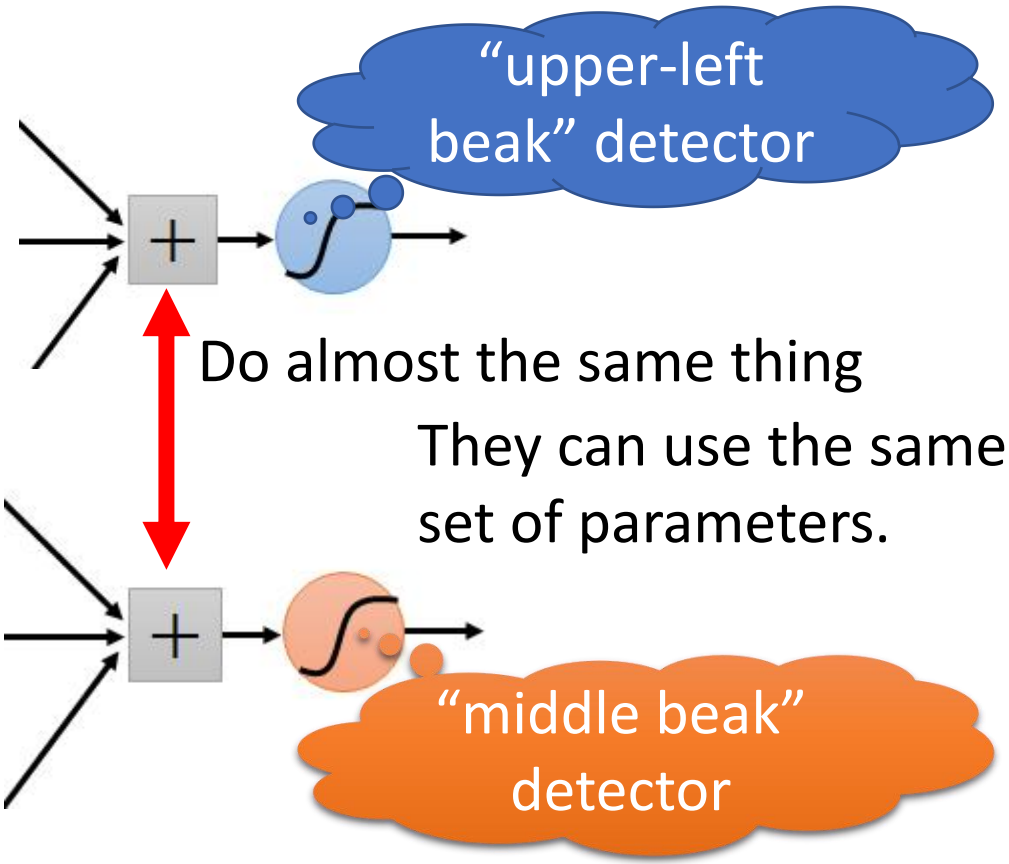
Connecting to small region with less parameters

参考自李宏毅老师课件

深度学习

- 卷积神经网络
  - The same patterns appear in different regions.





参考自李宏毅老师课件

卷积神经网络

- Subsampling the pixels will not change the object

bird

subsampling

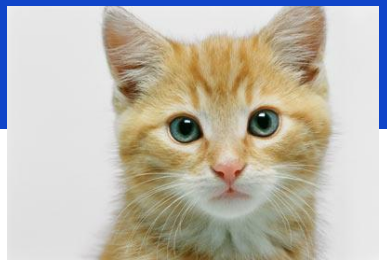
bird

We can subsample the pixels to make image smaller

→ Less parameters for the network to process the image

参考自李宏毅老师课件

The whole CNN



Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

Property 3

- Subsampling the pixels will not change the object

Convolution

Max Pooling

Convolution

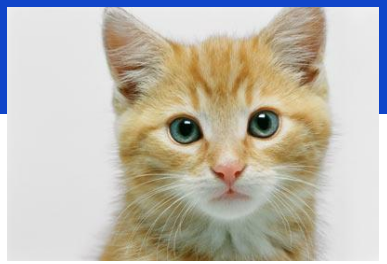
Max Pooling

Can repeat many times

Flatten

参考自李宏毅老师课件

The whole CNN



Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

cat dog ……

参考自李宏毅老师课件

深度学习

理解深度学习

- “特征工程” VS “特征学习”或者 “表示学习”
- 特征工程由人类专家根据现实任务来设计，特征提取与识别是单独的两个阶段；

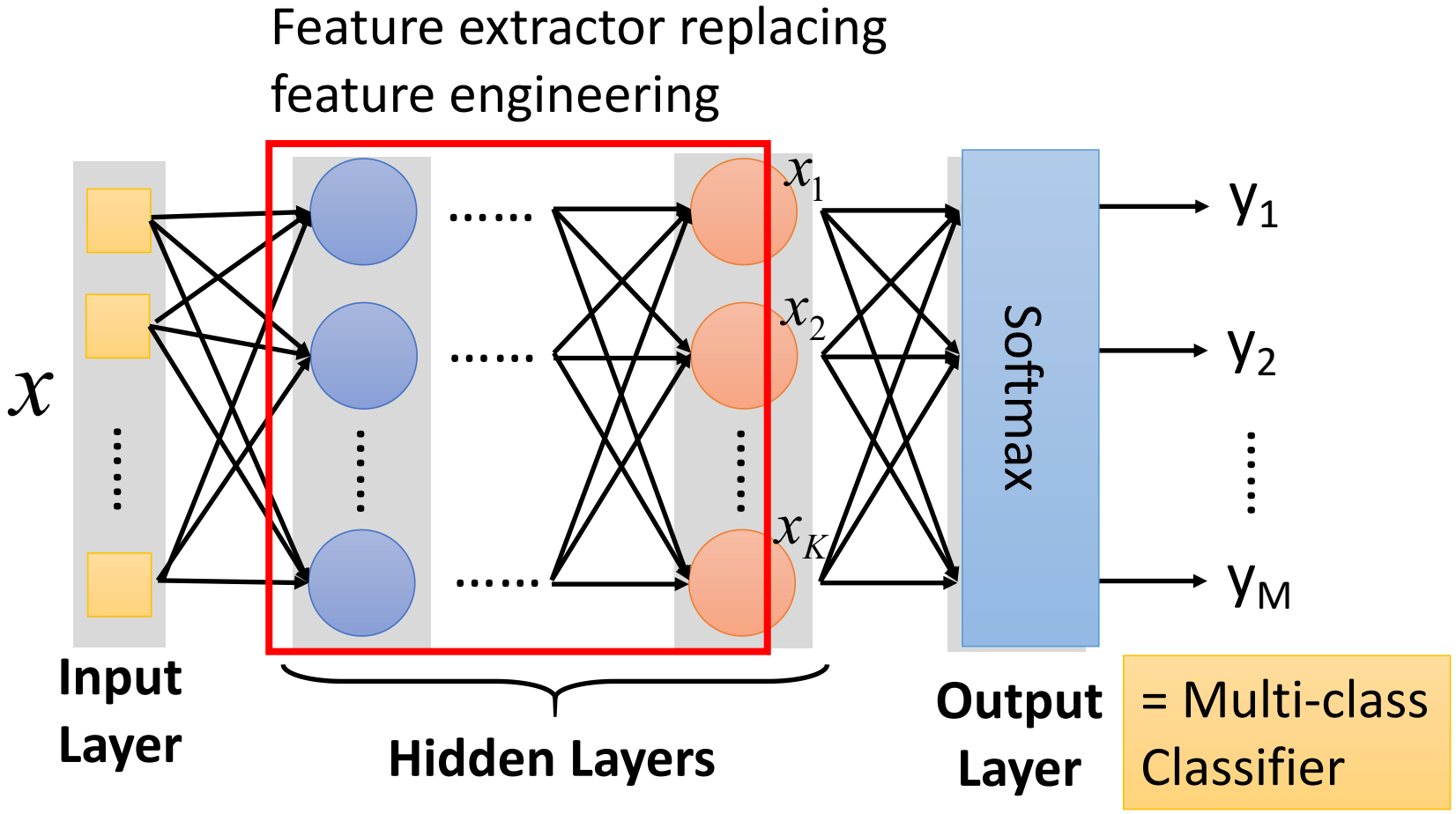
手工设计特征 → 分类识别

- 特征学习通过深度学习技术自动产生有益于分类的特征，是一个端到端的学习框架.

学习生成特征 → 分类识别

深度学习

- 理解深度学习



参考自李宏毅老师课件

Thanks