

人工神经网络漫谈

M-P 模型、感知器和 BP 网络

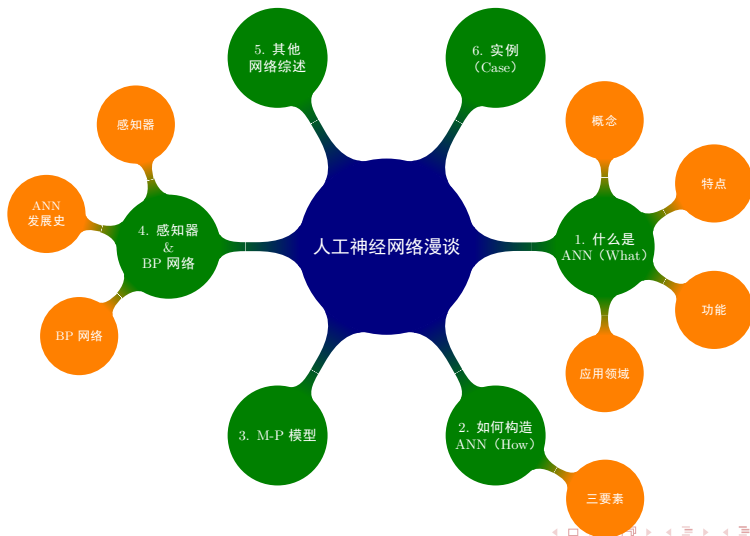
潘伟洲¹ 陈振洲²

¹Master Candidate
School of Computer
South China Normal University

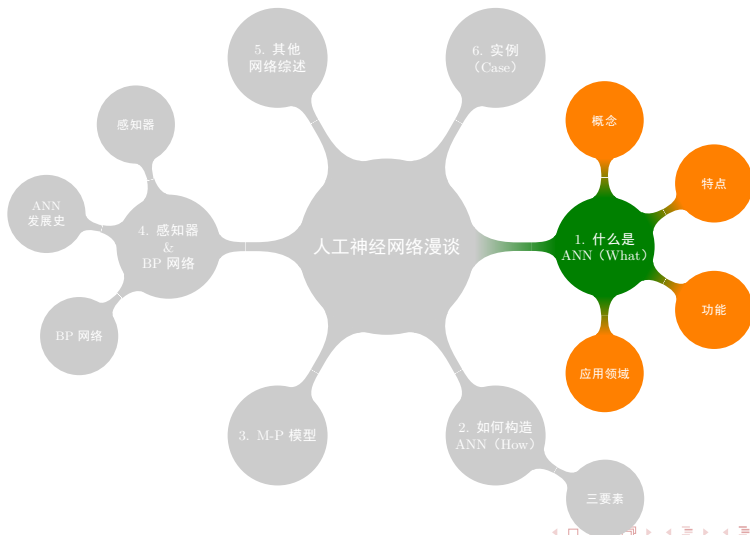
²Ph.D., Lecturer
School of Computer
South China Normal Univertisy

April, 2012

心智图



什么是人工神经网络?



加里·卡斯帕罗夫 V.S. 超级计算机“深蓝”。

- 1996 4:2 卡斯帕罗夫胜
- 1997 2.5:3.5 “深蓝”胜

▶ 经典人机大战



然而对于一些问题，计算机却一直不如人类聪明

- 辨认人脸
- 开车
- 打篮球
- ...

只可意会，不可言传！

然而对于一些问题，计算机却一直不如人类聪明

- 辨认人脸
- 开车
- 打篮球
- ...

只可意会，不可言传！

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的神经网络 1.4×10^{11} 个神经细胞
- 处理的信息：模拟量和离散脉冲
- 处理方式：并行
- 擅长联想、创造

电脑：冯 · 诺依曼方式

- 存储器和处理器相互独立
- 处理的信息：离散的二进制数和二值逻辑形式
- 处理方式：串行(取址、译码、执行、存储)
- 只擅长数值和逻辑运算

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的神经网络 1.4×10^{11} 个神经细胞
- 处理的信息：模拟量和离散脉冲
- 处理方式：并行
- 擅长联想、创造

电脑：冯 · 诺依曼方式

- 存储器和处理器相互独立
- 处理的信息：离散的二进制数和二值逻辑形式
- 处理方式：串行(取址、译码、执行、存储)
- 只擅长数值和逻辑运算

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

人脑与电脑的机制区别

人脑：神经网络

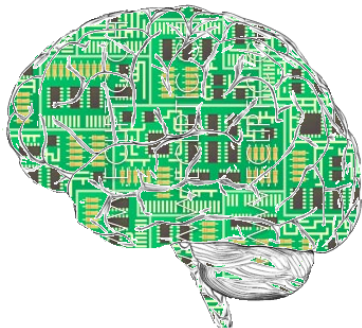
- 规模庞大、结构精细的**神经网络** 1.4×10^{11} 个神经细胞
- 处理的信息：**模拟量**和离散脉冲
- 处理方式：**并行**
- 擅长**联想、创造**

电脑：冯 · 诺依曼方式

- 存储器和处理器相互**独立**
- 处理的信息：**离散**的二进制数和二值逻辑形式
- 处理方式：**串行**(取址、译码、执行、存储)
- 只擅长**数值和逻辑运算**

什么是人工神经网络

- 在对人脑神经网络的基本认识上，用数理方法从信息处理的角度对人脑神经网络进行**抽象**，并建立某种**简化模型**。
- 不是人脑生物神经网络的真实写照，而只是对它的**简化、抽象与模拟**。
- 旨在**模仿**人脑结构及其功能的信息处理系统。



基本特点

结构特点	性能特点	能力特征
信息处理的并行性 信息存储的分布性 信息处理单元的互连性 结构的可塑性	高度的非线性 良好的容错性 计算的非精确性	自学习 自组织 自适应性

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

基本功能

- 联想记忆
 - 自联想记忆
 - 异联想记忆
- 非线性映射
- 分类与识别
- 优化计算
- 知识处理

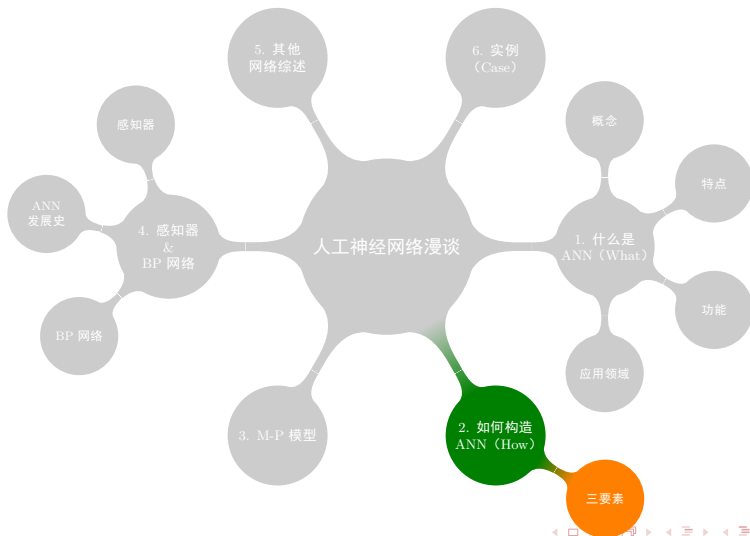
应用领域

神经网络学习方法对于逼近实数值、离散值或向量值的目标函数提供了一种健壮性很强的方法。对于某些类型的问题，如学习解释复杂的现实世界中的传感器数据，人工神经网络是目前知道的最有效的学习方法。

领域	应用举例
信息处理领域	信号处理、模式识别、数据压缩
自动化领域	系统辨识、自动控制、智能检测
工程领域	汽车工程、军事工程、化学工程、水利工程
医学领域	检测数据分析、生物活性研究、医学专家系统
经济领域	预测评估、信贷分析
...	...

问题

如何设计 ANN?



问题

如何设计 ANN?

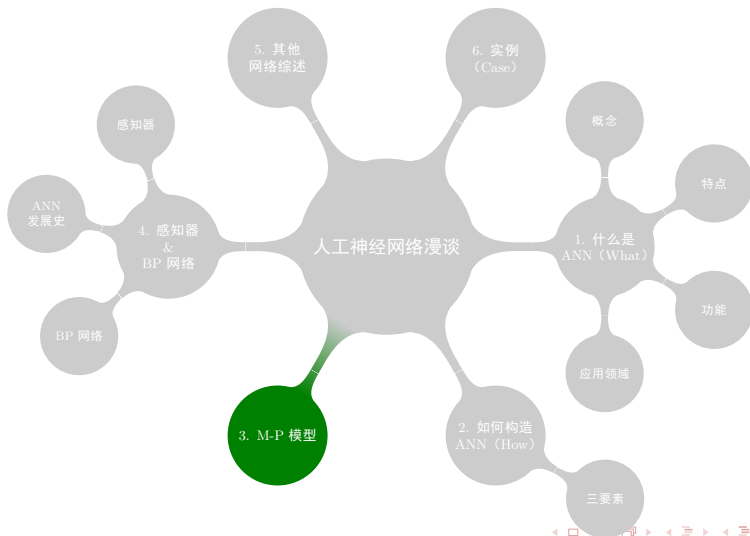
- ① 构成神经网络的最小单位是什么?
- ② 这些小单位怎么构造成神经网络?
 - 网络拓扑结构
- ③ 神经元之间如何进行信息传递?
 - 传递函数
- ④ 怎么训练这个神经网络使它学会我们要的某些本领?
 - 训练算法, 或者叫学习规则 (最终目的就是获得适用的权值)

ANN 三要素



Figure 1: ANN 三要素

M-P 模型



M-P 模型

从最简单开始——M-P 模型

- M-P 模型就是对一个生物神经元的建模
- 它是心理学家 W.McCulloch & 数学家 W.Pitts 于 1943 年提出的模型 (McCulloch-Pitts)

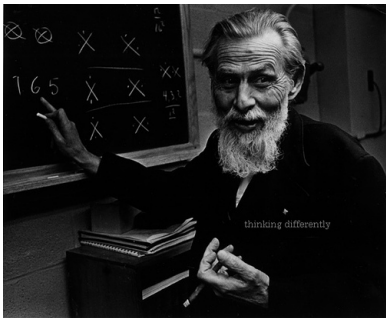


Figure 2: Warren S. McCulloch



Figure 3: Walter Pitts

生物神经元的结构

在谈 M-P 模型的内容之前，我们先得了解一下人脑中的神经元的结构，然后再研究 M-P 对人脑的神经元是如何建模的。

- 神经元是脑组织的基本单元
- 人脑中大约包含 1.4×10^{11} 个神经元
- 每个神经元与大约 $10^3 \sim 10^5$ 个其他神经元相连接

生物神经元的结构

Figure 4是一张生物神经元的简化示意图。

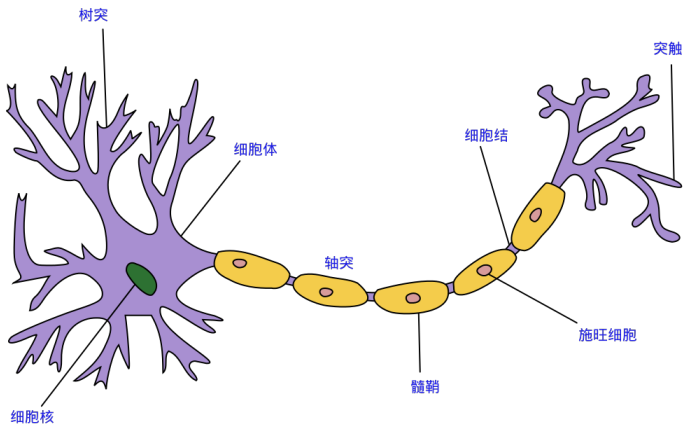


Figure 4: 生物神经元简化示意图

生物神经元的结构

神经元在结构上由**细胞体**、**树突**、**轴突**和**突触**4 部分组成。

① 细胞体

神经元的主体，由**细胞核**、**细胞质**和**细胞膜**3 部分组成。细胞体的外部是细胞膜，将膜内外细胞液分开。由于细胞膜对细胞液中的不同离子具有不同的通透性，这使得膜内外存在着离子浓度差，从而出现内负外正的静息电位。这种电位差称为**膜电位**。

② 树突

从细胞体向外延伸出许多突起神经纤维。负责接收来自其他神经元的输入信号，相当于细胞体的**输入端 (input)**。

③ 轴突

由细胞体伸出的最长的一条突起称为轴突。轴突比树突长而细。轴突也叫神经纤维，末端处有很多细的分支称为神经末梢，每一条神经末梢可以向四面八方传出信号，相当于细胞体的**输出端 (output)**。

④ 突触

一个神经元通过其轴突的神经末梢和另一个神经元的细胞体或树突进行**通信连接**，称为突触。

生物神经元的结构

突触使神经细胞的膜电位发生变化，且电位的变化是可以**累加**的，单个神经元可以与多达上千个其他神经元的轴突末梢形成突触连接，接受从各个轴突传来的**脉冲**输入。这些输入可到达神经元的不同部位，输入部位不同，对神经元影响的**权重**也不同。输入部位不同，该神经细胞**膜电位**是它所有突触产生的**电位总和**，当该神经细胞的膜电位升高到超过一个**阈值**时，就会产生一个**脉冲**，从而总和的膜电位直接影响该神经细胞兴奋发放的脉冲数。

神经元的信息是宽度和幅度都相同的**脉冲串**，若某个神经细胞兴奋，其轴突输出的脉冲串的频率就高；若某个神经细胞抑制，其轴突输出的脉冲串的频率就低，甚至无脉冲输出。因此，突触可以分为**兴奋性**和**抑制性**两种，兴奋性的突触可能引起下一个神经细胞兴奋，抑制性的突触使下一个神经细胞抑制。脉冲的传递是**正向**的，不允许逆向传播。另外，突触传递信息需要一定的**延迟**。

比喻 1：水桶模型

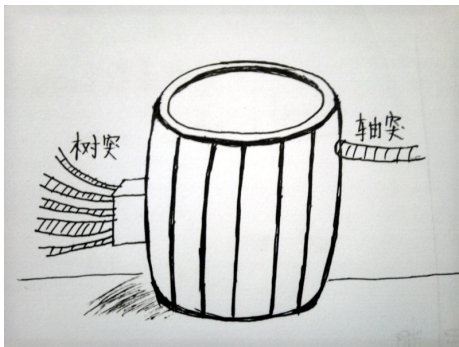


Figure 5: 水桶与神经元

我的一个比喻：可以把一个神经元想象成一个水桶，这个水桶侧边接着很多条水管（**神经末梢**），水管既可以将桶里的水输出去（**抑制性**），也可以将其他水桶的水输进来（**兴奋性**）。当桶里的水达到一个高度（**阈值**）时，就会通过另一条管子（**轴突**）将水输送出去。由于水管的粗细不同，对桶里的水的影响程度（**权重**）也不同。水管对水桶里的水位的改变（**膜电位**）自然就是这些水管输水量的累加。

比喻 II：选餐馆

我该去哪间餐馆吃饭?



M-P 模型

按照生物神经元，我们建立 **M-P 模型**。为了使得建模更加简单，以便于进行形式化表达，我们忽略时间整合作用、不应期等复杂因素，并把神经元的突触时延和强度当成常数。Figure 6就是一个 M-P 模型的示意图。

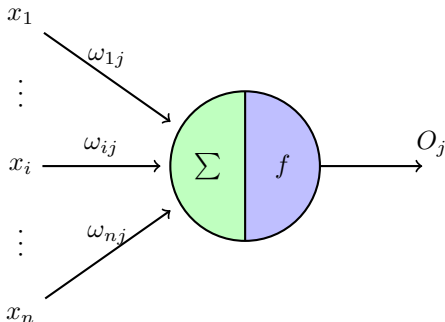


Figure 6: M-P 模型

M-P 模型

参照生物神经元，我们定义以下几个重要的参数：

Table 1: M-P 模型几个重要的参数

生物神经元	神经元	输入信号	权值	输出	总和	膜电位	阈值
M-P 模型	j	X_i	ω_{ij}	O_j	\sum	$\sum_{i=1}^n \omega_{ij} X_i(t)$	T_j

对于某一个神经元 j ，它可能接受同时接受了许多个输入信号，用 χ_i 表示，前面说过，由于生物神经元具有不同的突触性质和突触强度（水管粗细不同），所以对神经元的影响不同，我们用权值 ω_{ij} 来表示，其正负模拟了生物神经元中突出的兴奋和抑制（进水和出水），其大小则代表了突出的不同连接强度。由于累加性，我们对全部输入信号进行累加整合，相当于生物神经元中的膜电位（水的变化总量），其值就为

$$net'_j(t) = \sum_{i=1}^n \omega_{ij} \chi_i(t) \quad (1)$$

M-P 模型

神经元激活与否（外接专用水管流出与否）取决于某一阈值电平（水位高度），即只有当其输入总和超过阈值 T_j 时，神经元才被激活而发放脉冲，否则神经元不会发生输出信号。整个过程可以用下面这个函数来表示：

$$o_j(t) = f\left[\sum_{i=1}^n \omega_{ij}\chi_i(t)\right] - T_j \quad (2)$$

如果 $\chi_0 = -1$, $\omega_{0j} = T_j$, 则有 $-T_j = \chi_0 \omega_{0j}$
 因此公式 (1) 和 (2) 还可以简化成公式 (3) 和 (4) 的形式:

$$net'_j = W_j^T X \quad (3)$$

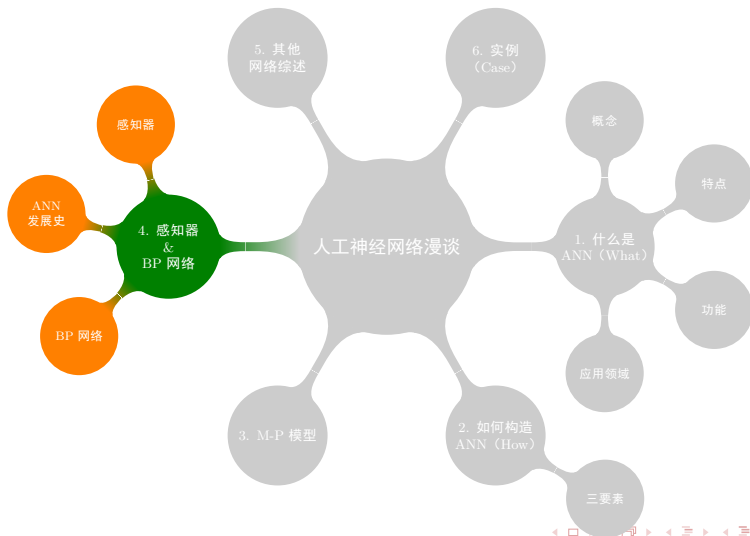
$$o_j = f(net_j) = f(W_j^T X) \quad (4)$$

M-P 模型的特点

由此我们可以得到总结出 M-P 模型的 6 个特点：

- ① 每个神经元都是一个**多输入单输出**的信息处理单元；
- ② 神经元输入分**兴奋性输入**和**抑制性输入**两种类型；
- ③ 神经元具有**空间整合特性**和**阈值特性**；
- ④ 神经元输入与输出间有固定的**时滞**，主要取决于突触延搁；
- ⑤ **忽略**时间整合作用和不应期；
- ⑥ 神经元本身是**非时变**的，即其突触时延和突触强度均为常数。

感知器 & BP 网络



什么是感知器

M-P 模型只是对单个神经元的建模，还不足以模拟人脑神经网络的功能。由这些人工神经元构建出来的网络，才能够具有学习、联想、记忆和模式识别的能力。

- 怎么连？
- 信号怎么传递？
- 怎么训练？

最简单的神经网络结构——单层感知器

1958 年，美国心理学家 Frank Rosenblatt 提出一种具有**单层计算单元**的神经网络，称为**感知器**(Perceptron)。

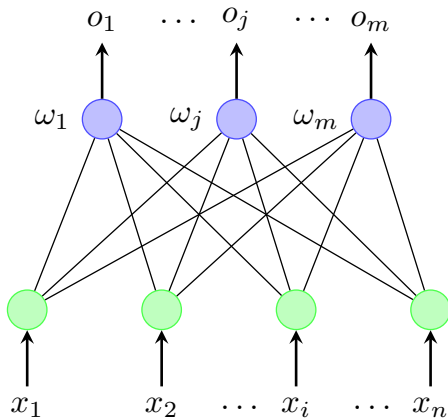


Figure 7: 单层感知器拓扑结构

单层感知器

- 输入加权和

$$net'_j = \sum_{i=1}^n \omega_{ij} \chi_i \quad (5)$$

- 学习规则：离散感知器学习规则（有导师学习）
- 传递函数：sgn 函数

$$o_j = \text{sgn}(net'_j - T_j) = \text{sgn}\left(\sum_{i=0}^n \omega_{ij} \chi_i\right) = \text{sgn}(W_j^T X) \quad (6)$$

- 公式 (6) 可以进一步表达为：

$$o_j = \begin{cases} 1 & W_j^T X > 0 \\ -1 & W_j^T X < 0 \end{cases} \quad (7)$$

风中烛——单层感知器的局限性

单层感知器仅对线性问题具有分类能力。

线性问题：简单来讲，就是用一条直线可分的图形。比如：

- 逻辑“与”
- 逻辑“或”

我们可以用一条直线来分隔 0 和 1。

感知器

逻辑“与”的真值表及二维样本分类图

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

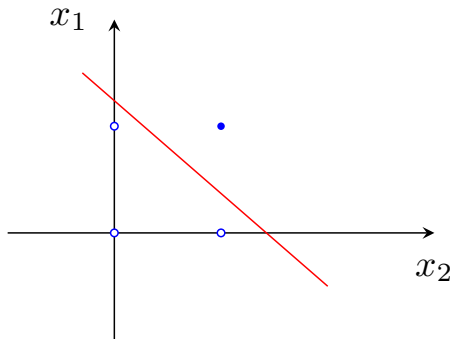


Figure 8: 逻辑“与”的二维样本分类图

逻辑“或”的真值表及二维样本分类图

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

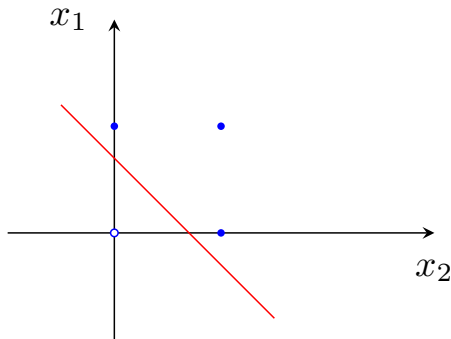


Figure 9: 逻辑“或”的二维样本分类图

为什么感知器就可以解决线性问题呢？这是由它的传递函数决定的。这里以两个输入分量 x_1 和 x_2 组成的二维空间为例，此时节点 j 的输出为

$$o_j = \begin{cases} 1 & \omega_{1j}x_1 + \omega_{2j}x_2 - T_j > 0 \\ -1 & \omega_{1j}x_1 + \omega_{2j}x_2 - T_j < 0 \end{cases}$$

所以，方程

$$\omega_{1j}x_1 + \omega_{2j}x_2 - T_j = 0 \quad (8)$$

确定的直线就是二维输入样本空间上的一条分界线。

“异或” 的真值表及二维样本分类图

如果要用它来处理非线性的问题，单层感知器网就无能为力了。例如下面的“异或”，就无法用一条直线来分割开来，因此单层感知器网就无法实现“异或”的功能。

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

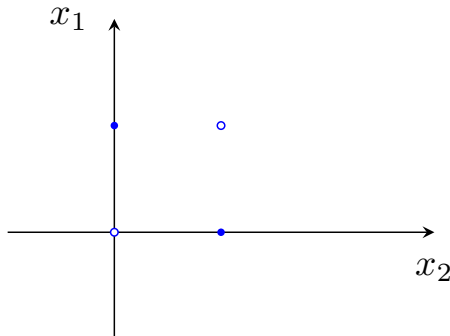


Figure 10: “异或” 的二维样本分类图

多层感知器

所谓多层感知器，就是在输入层和输出层之间加入隐层，，以形成能够将样本正确分类的凸域。多层感知器的拓扑结构如 Figure11所示。

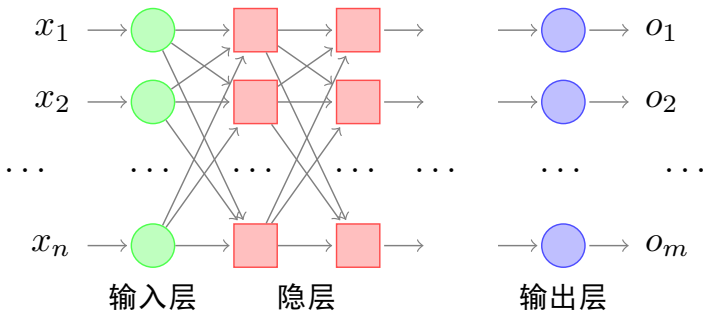






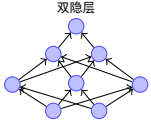




Figure 11: 多层感知器的拓扑结构

多层感知器

- 我们可以比较一下单层感知器和多层感知器的分类能力：

Table 2: 不同隐层数感知器的分类能力

结构	决策区域类型	区域形状	异或问题
无隐层 	由一超平面分成两个		
单隐层 	开凸区域或闭凸区域		
双隐层 	任意形状（其复杂度由单元数目确定）		

心有余而力不足——多层感知器的瓶颈

Kolmogorov 理论指出：**双隐层感知器足以解决任何复杂的分类问题。**

1966 年 Minsky 和 Papert 《感知器》：**多层感知器能够求解非线性问题，但对隐层神经元的学习规则尚无所知。**权值调整量取决于感知器期望输出与实际输出之差，即

$$\Delta W_j(t) = \eta[d_j - o_j(t)]X \quad (9)$$

对于各隐层节点来说，不存在期望输出，因而该学习规则对隐层权值不适用。

柳暗花明又一村——ANN 研究的复苏和 BP 神经网络的诞生

尽管 ANN 的研究陷入了前所未有的低谷，但仍有为数不多的学者忍受住寂寞，坚持致力于 ANN 的研究。在长达 10 年的低潮时期之间，相继有一些开创性的研究成果被提出来，但还不足以激起人们对于 ANN 研究的热情。一直到上世纪 80 年代，两个璀璨的成果诞生了：1982 年美国加州理工学院的物理学家 John J.Hopfield 博士的 Hopfield 网络和 David E.Rumelhart 以及 James L.McCelland 研究小组发表的《并行分布式处理》。这两个成果重新激起了人们对 ANN 的研究兴趣，使人们对模仿脑信息处理的智能计算机的研究重新充满了希望。

BP 网络

- 1986 年，Rumerhart 和 McClland 《平行分布式处理》对具有非线性连续变换函数的多层感知器的**误差反传（Error Back Propagation, BP）算法**进行了详尽的分析，实现了 Minsky 关于多层网络的设想。
- 由于多层感知器网络的训练经常采用误差反向传播算法，人们也常把多层感知器直接称为 BP 网。

误差反传算法

- Back Propagation was created by **generalizing** the **Widrow-Hoff** learning rule to **multiple-layer** networks and **nonlinear** differentiable transfer function.
- 误差反传算法 (BP 算法、 δ 算法)
 - 信号的正向传播
 - 误差的反向传播
- BP 算法的基本思想是，学习过程由信号的正向传播与误差的反向传播两个过程组成。

误差反传算法

信号的正向传播

- 正向传播时，输入样本从输入层传入，经各隐层逐层处理后，传向输出层。
- 若输出层的实际输出与期望的输出（教师信号）不符，则转入误差的反向传播阶段。

误差的反向传播

- 将输出以某种形式通过隐层向输入层逐层反传，并将误差分摊给各层的所有单元，从而获得各层单元的误差信号，此误差信号即作为修正各单元权值的依据。

BP 网络

BP 算法的信号流向

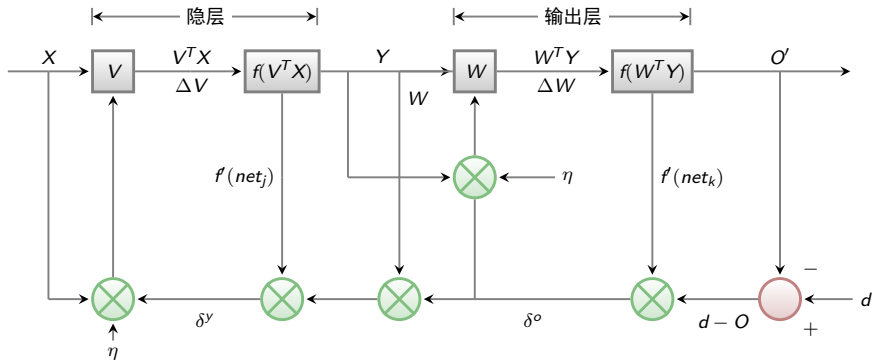


Figure 13: BP 算法的信号流向图

BP 网络的结构

- **多层感知器** [figure 11], 其中单隐层网络（三层感知器）的应用最为普遍。
- 三层：输入层、隐层、输出层。

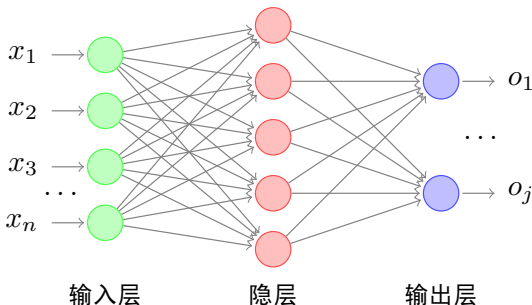


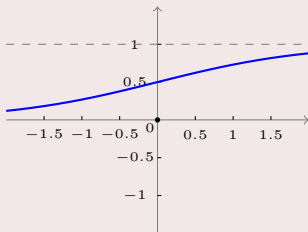
Figure 14: 三层 BP 网拓扑结构

传递函数——Sigmoid 函数

- 非线性变换函数，又称 S 型函数
- 特点：函数本身及其导数都是连续的

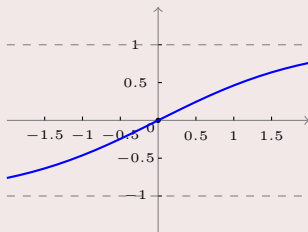
单极性 S 型函数

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



双极性 S 型函数

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



传统 BP 学习算法

以三层感知器 [figure 14] 为例。

1. 网络误差与权值调整

当网络输出与期望输出不等时，存在输出误差 E ，定义如下

$$\begin{aligned} E &= \frac{1}{2}(d - O)^2 \\ &= \frac{1}{2} \sum_{\kappa=1}^{\ell} (d_{\kappa} - o_{\kappa})^2 \end{aligned} \quad (10)$$

将以上误差定义式展开至隐层，有

$$\begin{aligned} E &= \frac{1}{2} \sum_{\kappa=1}^{\ell} [d_{\kappa} - f(\text{net}_{\kappa})]^2 \\ &= \frac{1}{2} \sum_{\kappa=1}^{\ell} [d_{\kappa} - f(\sum_{j=0}^m \omega_{j\kappa} y_j)]^2 \end{aligned} \quad (11)$$

传统 BP 学习算法

进一步展开至输入层，有

$$\begin{aligned} E &= \frac{1}{2} \sum_{\kappa=1}^{\ell} d_{\kappa} - f[\sum_{j=0}^m \omega_{j\kappa} f(\text{net}_j)]^2 \\ &= \frac{1}{2} \sum_{\kappa=1}^{\ell} d_{\kappa} - f[\sum_{j=0}^m \omega_{j\kappa} f(\sum_{i=0}^n v_{ij} \chi_i)]^2 \end{aligned} \quad (12)$$

由上式可以看出，网络输入误差是各层权值 $\omega_{j\kappa}$ 、 v_{ij} 的函数，因此调整权值可改变误差 E 。

显然，调整权值的原则是使误差不断减小，因此应使权值与误差的梯度下降成正比，即

$$\Delta \omega_{j\kappa} = -\eta \frac{\partial E}{\partial \omega_{j\kappa}} \quad j = 0, 1, 2, \dots, m; \quad \kappa = 1, 2, \dots, \ell \quad (13)$$

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}} \quad i = 0, 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (14)$$

传统 BP 学习算法

对于一般多层感知器，设共有 h 个隐层，按前向顺序各隐层节点数分别记为 m_1, m_2, \dots, m_h ，各隐层输出分别记为 y^1, y^2, \dots, y^h ，各层权值矩阵分别记为 $W^1, W^2, \dots, W^h, W^{h+1}$ ，则各层权值调整公式为

输出层

$$\Delta\omega_{j\kappa}^{h+1} = \eta \delta_{\kappa}^{h+1} y_j^h = \eta (d_{\kappa} - o_{\kappa}) o_{\kappa} (1 - o_{\kappa}) y_j^{\kappa} \quad j=0,1,2,\dots,m_h; \quad \kappa=1,2,\dots,\ell \quad (15)$$

第 h 隐层

$$\Delta\omega_{ij}^h = \eta \delta_j^h y_i^{h-1} = \eta \left(\sum_{\kappa=1}^{\ell} \delta_{\kappa}^o \omega_{j\kappa}^{h+1} y_j^{\kappa} (1 - y_j^{\kappa}) \right) y_i^{h-1} \quad i=0,1,2,\dots,m_{(h-1)}; \quad j=1,2,\dots,m_h \quad (16)$$

传统 BP 学习算法

按以上规律逐层类推，则第一隐层权值调整公式

$$\Delta\omega_{pq}^1 = \eta \delta_q^1 \chi_p = \eta (\sum_{r=1}^{m_2} \delta_r^2 \omega_{qr}^2) y_q^1 (1 - y_q^1) \chi_p \quad p=0,1,2,\dots,n; \quad j=1,2,\dots,m_1 \quad (17)$$

容易看出，BP 学习算法中，**各层权值调整公式形式上都是一样的**，均由 3 个因素决定，即：

- 学习率 η
- 本层输出的误差信号 δ
- 本层输入信号 Y

其中输入层误差信号与网络的期望输出与实际输出之差有关，直接反应了输出误差，而各隐层的误差信号与前面各层的误差信号有关，是**从输出层开始逐层反传过来的**。

传统 BP 学习算法

- 可以看出 BP 算法属于 δ 学习规则类，这类算法常被称为误差的梯度下降算法。
- δ 学习规则可以看成是 Widrow-Hoff(LMS) 学习规则的一般化 (generalize) 情况。LMS 学习规则与神经元采用的变换函数无关，因而不需要对变换函数求导， δ 学习规则则没有这个性质，要求变换函数可导。

改进的 BP 学习算法

标准的 BP 算法在应用中暴露出不少内在的缺陷：

- 极易形成局部最小而得不到全局最优；
- 学习效率低，收敛速度慢；
- 隐节点的选取缺乏理论指导；
- 训练时学习新样本有遗忘旧样本的趋势。

改进的 BP 学习算法

针对上述问题，国内外已提出不少有效的改进算法。

- 引入动量因子 (收敛速度);
- 引入自适应调节学习率 (收敛速度);
- 引入陡度因子 (收敛速度);
- 采用总体样本批训练法 (遗忘旧样本);
- 模拟退火算法、遗传算法等全局优化算法 (局部极小点);
- ...

总结：BP 网络的三要素



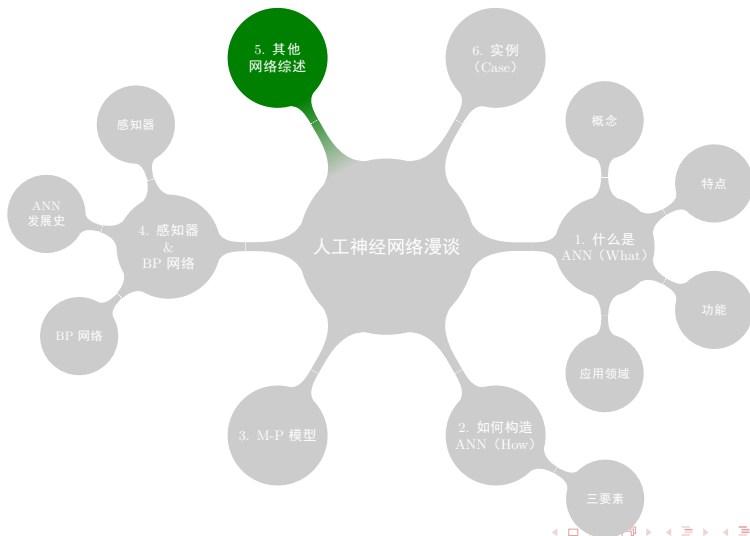
Figure 15: BP 网络的三要素

BP 网的功能

BP 网是至今为止应用最广泛的神经网络

- 非线性映射能力
- 泛化能力
- 容错能力

其他网络综述



人工神经网络概览 I

下表列出了神经网络发展过程中起过重要作用的十几种著名神经网络的情况，它也是神经网络发展史的一个缩影。

名称	提出者	诞生年	典型应用领域	弱点	特点
Perceptron (感知器)	Frank Rosenblatt (康奈尔大学)	1937	文字识别、声音识别、声纳信号识别、学习记忆问题研究。	不能识别复杂字形，对字的大小、平移和倾斜敏感。	最早的神经网络，已很少应用，有学习能力，只能进行线性分类。
Adaline (自适应线性单元) 和 Madaline (多个 Adaline 的组合网络)	Bernard Widrow (斯坦福大学)	1960-1962	雷达天线控制，自适应回波抵消、适应性调制解调、电话线中适应性补偿等。	要求输入 - 输出之间为线性关系。	学习能力较强，较早开始商业应用，Madaline 是 Adaline 的功能扩展。

人工神经网络概览 II

Avalanche (雪崩网)	S. Drossberg (波士顿大学)	1967	连续语音识别, 机器人手臂运动的教学指令	不易改变运动速度和插入运动。	
Cerellatron (小脑自动机)	D.Marr (麻省理工学院)	1969-1982	控制机器人的手臂运动	需要复杂的控制输入	类似于 Avalanche 网络, 能调和各种指令序列, 按需要缓慢地插入动作。
Back Propagation (误差反传网络)	P. Werbos (哈佛大学) David Rumelhart (斯坦福大学) James McClelland (斯坦福大学)	1974-1985	语音识别、工业过程控制、贷款信用评估、股票预测、自适应控制等	需要大量输入-输出数据, 训练时间长, 易陷入局部极小。	多层前馈网络, 采用最小均方差学习方式, 是应用最广泛的网络。

人工神经网络概览 III

Adaptive Resonance Theory (自适应共振理论 ART) 有 ART I、ART II 和 ART III 3 种类型	G.Carpenter 和 S.Grossberg (波士顿大学)	1976–1990	模式识别领域, 擅长识别复杂模式或未知的模式	受平移、旋转及尺度的影响。系统比较复杂、难以用硬件实现。	可以对任意多和任意复杂的二维模式进行自组织学习, ART I 用于二进制, ART II 用于连续信号。
Brain State in a Box (盒中脑 BSB 网络)	James Anderson (布朗大学)	1977	解释概念形成, 分类知识处理	只能作一次性决策, 无重复性共振。	具有最小均方差的单层自联想网络, 类似于双向联想记忆, 可对片段输入补全。

人工神经网络概览 IV

Neocognition (新认知机)	Fukushima K 福岛邦彦 (日本广播协会)	1978–1984	手写字母识别	需要大量加工单元和联系	多层结构化字符识别网络, 与输入模式的大小、平移和旋转无关, 能识别复杂字形。
Self-Organizing feature map (自组织特征映射网络)	Toevo Konhonen (芬兰赫尔辛基技术大学)	1980	语音识别, 机器人控制, 工业过程控制, 图像压缩, 专家系统等。	模式类型数据预先知道。	对输入样本自组织聚类, 映射样本空间的分布。
Hopfield 网络	John Hopfield (加州理工学院)	1982	求解 TSP 问题, 线性规划, 联想记忆和用于辨识。	无学习能力, 连接要对称, 权值要预先给定。	单层自联想网络, 可从有缺损或有噪声输入中恢复完整信息。

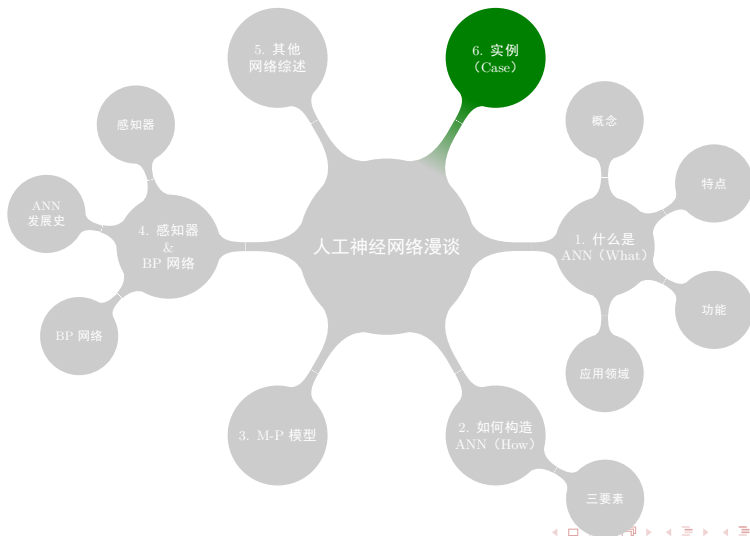
人工神经网络概览 V

Boltzman machine (玻耳兹曼机) Cauchy machine (柯西机)	J.Hinton (多伦多大学) T.Sejnowski (霍布金斯大学)	1985–1986	图像、声纳和雷达等模式识别。	玻耳兹曼机训练时间长，柯西机在某些统计分布下产生噪声。	一种采用随机学习算法的网络，可训练实现全局最优。
Bidirectional Associative Memory (BAM, 双向联想记忆网)	Bart Kosko (南加州大学)	1985–1988	内容寻址的联想记忆	存储的密度低，数据必须适应编码。	双向联想式单层网络，具有学习功能，简单易学。
Counter Proagation (CPN, 双向传播网)	Robert Hecht-Nielsen	1986	神经网络计算机，图像分析和统计分析。	需要大量处理单元和连接，需要高度准确。	一种在功能上作为统计最优化和概率密度函数分析的网络。

人工神经网络概览 VI

Radial Basis Functions (RBF, 径向基函数网络)	Broomhead Lowe	1988	非线性函数逼近, 时间序列分析, 模式识别、信息处理、图像处理、系统建模。	需要大量输入输出数据, 计算较复杂。	网络设计采用原理化方法, 有坚实的数学基础。
Support Vector Machine (SVM, 支持向量机)	Vapnik	1992–1998	模式分类, 非线性映射	训练时间长, 支持向量的选择较困难, 学习算法的推导较深奥。	在模式分类问题上能提供良好的泛化性能。

实例



怎么应用 ANN 解决问题？

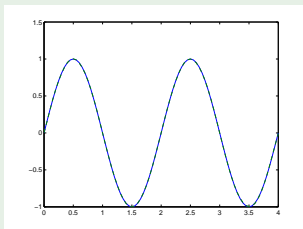
在使用 ANN 解决实际问题前，你应该先问自己下面几个问题：

- ① 什么时候可以用神经网络？
- ② 分类还是回归？
- ③ 确定的还是随机的？
- ④ 有监督还是无监督？
- ⑤ 在线还是离线？
- ⑥ PC 还是其他硬件设备（DSP 芯片）？

CASE I 曲线拟合

- 背景介绍：BP 网络采用非线性传递函数，相对线性网络具有更强的拟合能力，因此对于现实中遇到的各种复杂的函数波形可以进行很好的逼近。
- 实例要求：对一个具有一定采样率的正弦函数用 BP 网络来逼近其波形。

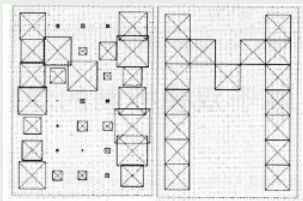
正弦信号



CASE II 特征识别

- 背景介绍：利用计算机进行模式识别是一项很有用的技术，尤其是利用机器来识别图形符号的特征。
- 实例要求：设计并训练一个 BP 网络，完成 26 个字母的图像识别。

三层 BP 网拓扑结构



Thank You!