



土木与建筑工程CAE

第七章 CAE系统的智能化

清华大学土木工程系

胡振中

huzhenzhong@tsinghua.edu.cn

Tsinghua University



第七章 CAE系统的智能化

7.1 智能系统与智能CAE

7.2 知识的获取表示与处理

7.3 知识库系统

7.4 基于人工神经元的知识处理技术

7.5 MATLAB神经网络工具箱



7.1 智能系统与智能CAE

- 问题的提出
 - 计算机长于处理数值计算（结构分析、计算）问题，短于处理模糊知识或不充分条件描述问题。在需要由经验或知识进行（方案选择、初始截面确定）决策的时候，常需要人为地进行较多的干预。
 - 用计算机出错问题
 - 输入数据错 — 智能查错
 - 计算模型错 — 智能建模
 - 计算结果错 — 智能检查



7.1 智能系统与智能CAE

- 提出CAD智能化问题
 - 现有CAD存在一定的限制和不足：
 - ✓ 不能识别设计进行中的变化；
 - ✓ 计算机综合处理能力不如分析处理；
 - ✓ 不能处理模糊知识或不充分条件描述的设计问题；
 - ✓ 将需要经验或知识决策的设计问题留给用户交互输入，故人为干预较多，因而设计质量在一定程度上依赖用户的经验和知识。
- 智能系统就是使计算机系统具有智能的特征，使其具有知识，能根据知识进行思考。



7.1 智能系统与智能CAE

- 智能系统的主要特征
 - 合理的物理结构—计算机硬件体系
 - 智能行为产生的物质基础（速度、内存等）
 - 完善的知识系统
 - 知识是智能体所具有的分析 and 解决问题的经验、策略、认识和原则等。
 - 知识是智能行为的基础。
 - 健全的思维机制
 - 智能体运用自身知识对信息加工的过程。
 - 思维能力即获取和运用知识的能力，即学习能力、理解能力、记忆能力、分析问题解决问题的能力。

智能系统是具有知识并能运用知识解决问题的实体系统。



7.1 智能系统与智能CAE

- 人工智能
 - 人工智能是使计算机具有人类行为的计算机科学的一个分支。
 - 研究用人工的方法和技术模仿、延伸和扩展计算机的智能，使计算机呈现出人类的智能行为，如学习、理解、分析、推理等认识问题和解决问题的特性。
 - 问题求解
 - 逻辑推理与定理证明
 - 自然语言处理
 - 机器学习
 - 专家系统
 - 机器人
 - 智能检索
 - 组合和调度问题
 - 系统与表达语言



7.1 智能系统与智能CAE

● 智能CAD

- 人工智能和CAD技术相结合的一门综合性学科。
- 尽可能采用自动化设计技术，使计算机参与方案决策、结构设计、性能分析、图形处理等设计全过程：
 - 完全面向对象的参数设计
 - 评价设计的完善性、合理性
 - 引导高层次的自动分析及设计
 - 评估及改善设计质量

● 智能CAE

- 把人工智能的思想、方法和技术引入到CAE系统中。
- 模拟人类的思维方法、进行分析、判断、推理，完成规划、设计、施工、维护的工程建设全过程。
- 建立不同性质类型工作的紧密耦合，对工程建设全过程提供一体化的计算机支持。



7.2 知识的获取表示与处理

- 知识的定义
- 知识的类型
- 知识的获取
- 知识的表示
- 知识的处理



知识的定义

- 人工智能技术将知识定义为**人类专家在特定专业领域的经验性知识**。
 - 规则和窍门 (Know-how) 等；
 - 特点：非结构化，无法用结构化的模型（数学模型）来描述。
- 从认识论的角度，**知识为人类对于客观事物规律性的认识**。
 - 数据：数值和符号，通常只是事物的名称；
 - 信息：通过数据之间的某种联系，揭示有意义的概念；
 - 知识：通过一些信息的联系，揭示出事物的规律性。
 - **知识是关于信息的信息**



知识的类型

- **叙述型**：描述事物的属性、数量、事实、状态、概念、定义等。
- **过程型**：描述事物的属性变化、状态转换、数据演算的行动规则和操作等方法。
- **控制型**：描述如何选择相应的操作、演算和行动的比较、判断、管理和决策的知识。



知识的获取（机器感知）

- 传统的知识录入
- 规则化的推导
- 研究如何直接或间接获取（自动或半自动）知识，输入自然信息（文字、图象、声音、语言、景物）
 - 机器视觉：文字、图象识别、景物分析。
 - 机器听觉：声音识别、语言理解。
 - 机器触觉：力感觉、平衡感觉等（数据手套）。



知识的表示

- 知识表示的层次
 - **对象级知识**：与问题求解有直接关系的知识；
 - **元级知识**：与对象级知识性质和使用方法有关的知识。
- 知识表示
 - 为描述世界所作的一组符号，是知识的符号化过程，以便把人类知识表示成计算机能处理的知识结构。
- 知识表示方法
 - 指用计算机表示知识的可行性、有效性的一般方法，是用数据结构和控制结构的统一体，既考虑知识的储存，又考虑知识的使用。



知识的表示

- 知识表示方法的性能
 - 充分表达：具备确切表达有关领域中各种知识的能力；
 - 有效推理：能够与高效的推理功能密切地结合；
 - 便于管理：
 - 便于实现模块化；
 - 便于检测矛盾知识和冗余知识；
 - 便于知识更新；
 - 便于知识库维护。
 - 易于理解：使知识的表示结构具有透明性，对于知识的输入、错误的检测、解释功能的实现至关重要。
- 常用的知识表示方法
 - 逻辑
 - 语义网络
 - 框架
 - 产生式规则



知识的表示：逻辑

- 逻辑
 - 逻辑是表达人类思维和推理的最成功的方法，表现方法和人类自然语言接近，易被人们所接受，能被计算机精确处理。
 - 逻辑方法的演绎结果在一定范围内保持正确，从现有事实推导出新事实可以用计算机实现。
 - 知识的逻辑表示是以陈述性方法为主的知识表示方法。主要是运用命题逻辑、谓词逻辑等知识来描述事实。



知识的表示：逻辑

- 逻辑表示的基本组成：
 - 谓词、逻辑符号、圆括弧、方括弧、花括弧。
 - 它们之间用逗号隔开，以表示域内的关系，组成原子公式。
CHINESE (LI MING) 李明是中国人。
MARRIED [brother(LI), sister (WANG)] 李的哥哥与王的姐姐已结婚。
- 逻辑符号：
 - ^与、∨或、→隐含、¬非、≡等价.....
- 用逻辑进行推理：
 - 在知识库中寻找合适的知识，进行模式匹配，而后提出新的事实。



知识的表示：逻辑

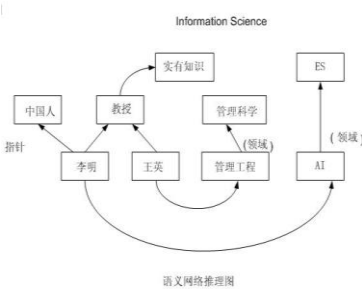
- 语言
 - PROLOG（逻辑型语言）、LISP语言（函数型语言）
- 特点
 - 知识表达自然、准确、灵活，易于模块化，相应的推理方法采用归结原理，严格、完备、通用。
 - 能表达的知识较简单，无法方便地描述复杂的知识结构，推理方法易产生组合爆炸。
 - 推理效率低，对工程问题不很适合。



知识的表示：语义网络

- 语义网络
 - 由表示实体、概念、事件等的结点和表示连接结点之间的关系的弧所组成的有向图。

Information Science



语义网络推理图



知识的表示：语义网络

- 该图表示了以下8个事实：
 - 李明是中国人。
 - 人工智能是信息科学的一个领域。
 - 王英是教授。
 - 管理工程是管理科学的一个领域。
 - 教授是有知识的。
 - 李明是教授。
 - 李明是人工智能教授。
 - 王英是管理工程教授。
- 在语义网络中，人们考虑某个问题时，并不需要牵涉庞大的知识库，而只要相关知识即可。



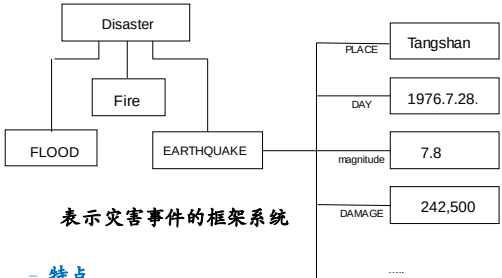
知识的表示：语义网络

- 特点
 - **知识的适合表达**：能把实体的结构、属性、及实体间的因果联系显示、简明地表达出来，便于以联想方式实现系统解释。
 - **知识结构化组织**：与一个概念相关的属性和联系被组织在一个节点中，便于被访问和学习。
 - **自然性**：知识表示直观，易于理解，适于与领域专家沟通，符合人类的思维习惯。
 - **非有效性**：推理过程中有时不能区分物体的“类”和“个体”节点。
 - **非清晰性**：节点之间的简单线状、树状、网状联系，给知识存储、修改和检索造成困难。



知识的表示：框架

- 框架
 - 框架理论
 - 当人们对周围事物进行观察时，外界的事物、发生的事件，环境的多次刺激在人们脑中形成一个个的概念，这些概念在人们头脑中用图形记忆。这些图形在计算机中可以表示成框架或称画面。
 - 当人们遇到一个新情况时，人们并不从头分析情况，而是把老框架拿出来和新情况作比较，进行联想，作出推理判断。
 - 框架结构
 - 框架是由若干节点和关系构成的网络。
 - 是一个多叉树结构。



- 特点
 - 框架与语义网络在本质上是相通的，同属知识的结构化表示方法。
 - 框架的逻辑推理，更接近人的认知和思维，适于表达复杂对象。
 - 可表达静态知识，具有良好的模块化，适于计算机处理。



知识的表示：产生式规则

- 产生式规则
 - 以产生式规则作为基本知识表达方式的知识库系统。在产生式系统中，可以用正向或逆向推理作为基本的求解策略。
 - 产生式规则的表达形式

```
IF      前提1 AND (OR)
      前提2 AND (OR)
      ...
      前提n
THEN    结论
```



知识的表示：产生式规则

- 产生式系统
 - **知识库**：事实及其相互关系
 - **产生式规则**：规则的集合
 - **解释器**：控制系统的运行。
- 特点
 - **有效表达过程知识**：即关于操作和行动的知识，一般以“怎么做”的形式表达，“IF-THEN”规则可表达一个过程的操作。
 - **自然的表达知识**：符合人的思维，便于知识获取，易于理解。
 - **规则的独立**：规则库中每一条规则为一个基本知识单元，规则之间不互相调用，使知识库易于修改和完善。
 - **易于实现试探性推理**。
 - **不适于表达结构性知识**。
 - **解释能力局限性**：只反映条件与结论的关系，而不能反映该条件为何能产生其结论。
 - **大型知识库推理效率低**。



知识的处理

- 研究在机器中如何存储、组织与管理知识，如何进行知识推理和问题求解。
- 三段论式推理
 - 根据一个或一些判断得出另一个判断的思维过程，推理所根据的判断，叫做**前提**。
 - 由前提得出的那个判断，叫做**结论**。
 - **三段论**：由三个性质判断组成，其中两个是前提，一个是结论。
例：
(1) 前提：所有的推理系统都是智能系统
(2) 前提：专家系统是推理系统
(3) 结论：专家系统是智能系统
 - 在智能系统中，推理机利用知识库的知识，按一定的推理策略去解决当前的问题。



知识的处理

● 归纳推理

- 由个别的事物或现象推出该类事物或现象的普遍性规律的推理。
- 常见的归纳推理法有简单枚举法、类比法、统计推理求因果五法（契合法、差异法、契合差异并用法、共变法与剩余法）

● 简单枚举法

- 由某类中已观察到的事物都有某属性，而没有观察到相反的事例，从而推出某类事物都有某属性，可靠性较差。
- 例，证明 $1^2 + 2^2 + \dots + n^2 = \frac{1}{6}n(n+1)(2n+1)$

用数学归纳法



知识的处理

● 类比推理

- 在两个或两类事物的许多属性上都相同的基础上，推出它们在其他属性上也相同。
- 类比法的基础是相似原理。
- 类比法的可靠程度取决于两个或两类事物的相同属性与推出的那个属性之间的相关程度，相关程度越高，类比法的可靠性就越大。



7.3 知识库系统

● 知识库系统

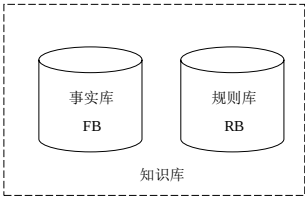
- 把人类具有的知识以一定形式表示后存入计算机，实现方便而有效地使用、管理大量的知识。
- 它是任何基于知识的智能系统的基础。
- 在知识库中，采用适当的知识表达技术，如产生式规则、语义网络、框架结构等，将知识表达为相应的符号结构，存储在计算机的存储设备中，把知识编排和组织起来。



产生式系统的知识库

● 特点

- 事实库 (Fact Base) 与规则库 (Rule Base) 相对独立，便于对它们分别进行增删、修改、更新。
- 事实库的内容可供所有规则进行访问，规则库的规则也可供所有事实使用。
- 规则库中各规则相对独立，不相互调用，不相互牵制



产生式系统的知识库

- 优点

- 利于知识库增删、修改，便于获取知识、灵活利用知识。

- 缺点

- 检索、查询效率低，对复杂问题会产生“组合爆炸”。

- 实际应用

- 与其他方法如语义网络结合。



知识库管理系统KBMS

● 知识库管理系统 KBMS

- 知识存入功能

- 知识增删、修改、补充、更新、积累等，以及学习功能。

- 知识取出管理功能

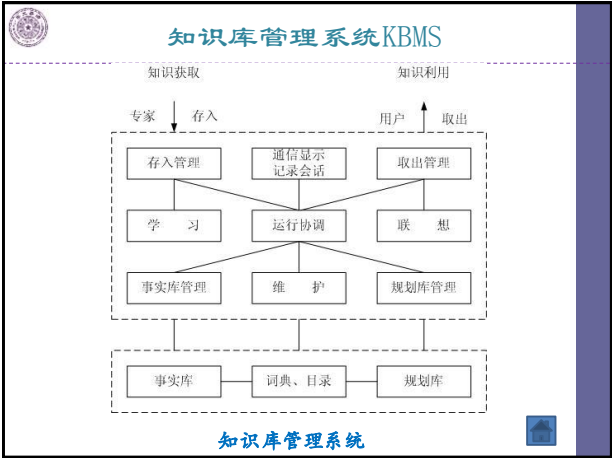
- 知识检索、查询、调用、关系推理和联想等。

- 知识库维护功能

- 知识保密、矛盾性、冗余性、完整性的检查。

- 知识库运行管理功能

- 会话、通信、显示、记录管理，以及存取控制、管理协调、知识共享等。



7.4 基于神经网络的知识处理技术

- **人工神经网络 (Artificial Neural Network)**
 - 是以一种简单的方式从结构上来模拟人脑神经网络，从而实现模拟人脑思维活动的功能。
 - 是由大量类似于脑神经元的简单处理单元(即人工神经元)广泛相互连接而成的复杂网络系统。
 - 人脑约有 10^{11} 个脑细胞，每一个脑细胞是一个活动的信息处理单元，称为**神经元**。
 - 每个神经元由大约 10^5 个的传递信息路径与其他神经元相连，从而组成一个复杂网络，即人脑的神经系统，亦称为**神经网络**。

人工神经网络的发展历史

- 研究大脑工作始于1890年
- 1949. Donald Hebb. **"The Organization of Behavior"** 为重要的研究著作。提出了神经元之间突触联系强度可变的假设，为神经网络学习算法奠定了基础。
- 50年代计算机硬、软件发展，开辟了计算机仿真时代，神经网络的术语开始出现。
- IBM研究实验室的Nathaniel Rochester等人基于Hebb的研究，建立了神经网络的软件模拟，第一次尝试失败，后在Hebb等人帮助下进行了成功的改进，第一次在计算机上实现，奠定了该门学科的基础。

人工神经网络的发展历史

- 1959年Bernard Widrow和Marcian Hoff第一次用神经网络来处理实际问题，用作在适应性滤波器消除电话线路中的回音，这一技术目前仍在用。
- 受1969 Minsky等人在"Perceptrons" (感知器) 中的影响。(认为神经网络不能解决复杂非线性问题)，该领域冷清了十多年。
- 1982 John Hopfield提出可以解决一些复杂问题，此后该领域迅速发展，也推出各种神经网络软件 (EI每年收录6000篇论文)。

人工神经网络的应用

- **预测**
 - 股票市场、天气、销售、工业过程劳动力、经济指标等预测。
- **分析**
 - 光谱分析与解释、疾病诊断、石油探测。
 - 尿样结果分析、精神病诊断、雇员选择。
 - 投资市场分析、科学仪器故障分析。
- **分类**
 - 聚合物鉴定、云层分类、产品分级、图象识别。

人工神经网络的应用

- **优化**
 - 运动员训练优化、机器维修期优化。
 - 生物试验结果优化。
- **控制**
 - 质量控制、过程控制、聚变炉优化操纵。
- **其他**
 - 金属试验、语言教学、计算机辅助翻译。

目前，神经网络应用于越来越多的领域。



人工神经网络技术的特性

- 与一般的传统数据处理方法相比、神经网络技术有下述优越性：
 - 模糊的数据
 - 需要决定的模式特征不明确
 - 数据本身非线性
 - 随机数据或数据中含有较多的噪声。
- 人脑是一个高度非线性的动力学系统：
 - 每个神经元的结构和功能十分简单，但大量神经元构成的神经网络的活动行为，却异常复杂，**整体的活动性质不等于单个神经元活动的简单相加。**



人工神经网络技术的特性

- 非线性的基本特征表现：
 - 人脑存储的长期记忆信息通过**改变突触的效能**来实现，每一条信息记录在许多突触上，每一突触又同时记录许多不同的信息。
 - 人脑的操作是**大规模并行**的。在脑神经系统中，人脑的活动反映在每个神经元并行地操作，并且每一种神经元存储的信息也同时参与操作。当神经元被驱动时，信号在不同神经元之间经突触（连接弧）沟通。
- 基于人工神经网络知识处理的集约特征：
 - 知识的获取、表示和利用合为一体，都是**通过神经网络的学习训练实现的**，从而可以充分利用**样本性知识**；
 - 样本性知识相对容易获得，为解决人工智能的知识获取“瓶颈”问题提供了可行的途径。
 - 人工神经网络模型实现的是右半脑形象思维的特性，而符号模型实现的是左半脑逻辑思维的特性，二者具有互补作用，而不能简单取代。



人工神经元模型：神经元的构成



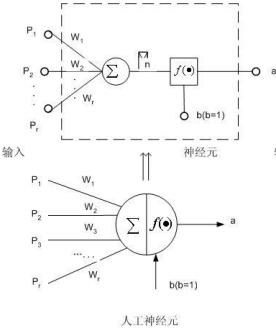
生物神经元

- 一个生物神经元的构成：
 - 细胞体。
 - 树突（输入端）：收集由其他神经元传来的信息。
 - 轴突（输出端）：传出从细胞体传来的信息。
 - 突触：两个神经细胞之间的相互接触点，决定了神经元之间相互作用的强弱。

- 一神经元的**树突经突触**与另一神经元的**轴突**相连
- 树突从突触接受信息，并引导到细胞体，激起神经元兴奋或抑制，从而决定了神经元的状态。
- 当神经元胞体内累积超过阈值时，神经元被驱动，沿轴突发送信号到其他神经元。



人工神经元模型：神经元的构成



- 人工神经元是对生物神经元主要功能和特性的数字抽象，一般为多输入/单输出的非线性元件。
 - $P_1, P_2, \dots, P_r$ 为输入；
 - $W_1, W_2, \dots, W_r$ 为权值：
 - 权重系数: 兴奋与抑制的强度
 - $f(\cdot)$ 为传递函数（激活函数）：
 - 神经元对来自四面八方的输入进行总和和 Σ 作用。
 - 并根据传递函数进行计算，确定是否输出信息。

人工神经元



人工神经元模型：神经元的构成

- 权值系数：表示输入的重要程度
 - 人工神经元的每个输入都有一个给定的权值，权值因子直接影响输入的效果，当多个输入结合产生神经脉冲时，有些输入就比其他输入更重要，起更大的作用。
- 激活函数的输入：

激活函数 $f(\cdot)$ 的输入： $P_j (j=1,2,\dots,r)$

通过与它相乘的权值分量 $W_j (j=1,2,\dots,r)$ 相连，以 $\sum_{j=1}^r W_j P_j$ 的形式求和后，形成激活函数：

$$[W_1 \ W_2 \ \dots \ W_r] \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_r \end{bmatrix} \quad f(\cdot) \text{ 的输入。}$$



人工神经元模型：神经元的构成

- 偏差：表示神经元内部其他因素的影响
 - 神经元输出除受输入信号的影响外，同时也受到**神经元内部其他因素**的影响，所以在人工神经元的建摸中，常常还加有一个额外输入信号，称为**偏差** (bias)，有时也称为**阈值**或门限值，b常为1。
- 神经元的输出矢量：
 - 权值 W_j 和输入 P_j 的矩阵形式可表示为： $W = [W_1 \ W_2 \ \dots \ W_r] \quad P = [P_1 \ P_2 \ \dots \ P_r]^T$

∴ 神经元模型的输出矢量可表示为：

$$A = f(W * P + b) = f\left(\sum_{j=1}^r W_j P_j + b\right)$$



人工神经元模型：激活函数

● 激活函数

- 激活函数是一个**神经网络的核心**。神经网络解决问题的能力与功效除了与网络结构有关，在很大程度上取决于该网络所采用的激活函数。
- **激活函数的作用：**
 - 控制输入对输出的激活作用（是否输出、输出什么）
 - 对输入、输出进行函数转换
 - 将可能无限域的输入变换或指定的有限范围内的输出

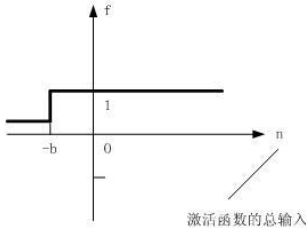


人工神经元模型：激活函数

(1) 阈值型（硬限制型）

- 将任意输入转化为0或1的输出。

$$A = f(W * P + b) = \begin{cases} 1 & W * P + b \geq 0 \\ 0 & W * P + b < 0 \end{cases}$$

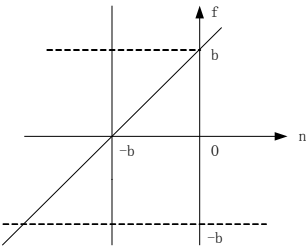


人工神经元模型：激活函数

(2) 线性型

- 使网络的输出等于加权输入和加上偏差。

$$A = f(W * P + b) = W * P + b$$

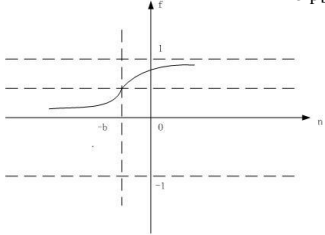


人工神经元模型：激活函数

(3) S型 (Sigmoid)

- 将任意输入值压缩到 (0, 1) 的范围内。

1) 对数S型激活函数:
$$f = \frac{1}{1 + \exp[-(n+b)]} \left(\frac{1}{1 + e^{-(n+b)}} \right)$$



n 指激活函数的总输入

$$\sum_{j=1}^n W_j P_j$$

- ① 连续单调上升函数
- ② 极限值为0 ($n \rightarrow -\infty$)
极限值为1 ($n \rightarrow +\infty$)
- ③ 可直接表达权重因子的抑制激活作用

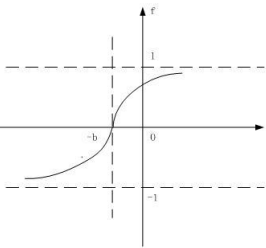
带有偏差的对数S型激活函数



人工神经元模型：激活函数

2) 双曲正切S型激活函数

$$f = \frac{1 - \exp[-2(n+b)]}{1 + \exp[-2(n+b)]} \quad n \rightarrow \infty \quad f = \frac{1 - \frac{1}{e^{2(n+b)}}}{1 + \frac{1}{e^{2(n+b)}}} = 1$$



$$\rightarrow -\infty \quad f = \frac{e^{2(n+b)} - 1}{e^{2(n+b)} + 1} = -1$$

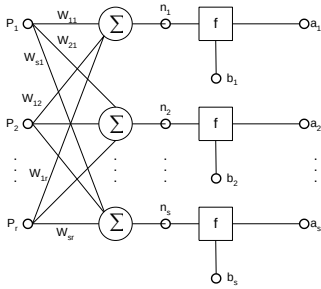
一般，称一个神经网络是线性或非线性是由网络的神经元中所具有的**激活函数的线性或非线性**来决定的。

双曲正切S型激活函数



单层神经网络模型

- 将两个或更多的简单的神经元并联，使每个神经元具有相同的输入矢量，即可组成一个神经层，其中每个神经元产生一个输出。



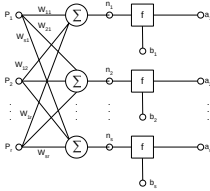


单层神经网络模型

- 输入矢量P的每个元素 $P_j (j=1,2,\dots,r)$
通过权矩阵W与每个输出神经元相连；

每个神经元通过一个求和符号，在与输入矢量进行加权求和运算后，形成激活函数的输入矢量，并经过激活函数 $f(\cdot)$ 作用后得到输出矢量A:

$$A_{s \times 1} = F(W_{s \times r} * P_{r \times 1} + B_{s \times 1})$$



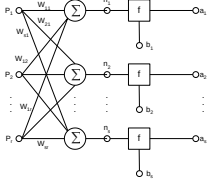
单层神经网络模型

S 神经元的个数

一般地，输入分量数与神经元个数S不相等，即 $r \neq S$

网络权矩阵

$$W_{s \times r} = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1r} \\ W_{21} & W_{22} & \dots & W_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ W_{s1} & W_{s2} & \dots & W_{sr} \end{bmatrix}$$



行 (神经元个数s)

列 (输入矢量个数r)

W_{12} : 来自第2个输入元素到第一个神经元之间的联接权值。

W_{ij} : 来自第j个输入元素到第i个神经元之间的联接权值。



单层神经网络模型

当有q组r个输入元素作为网络的输入时，输入矢量P成为一个 $r \times q$ 的矩阵

$$P_{r \times q} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1q} \\ P_{21} & P_{22} & \dots & P_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ P_{r1} & P_{r2} & \dots & P_{rq} \end{bmatrix}$$

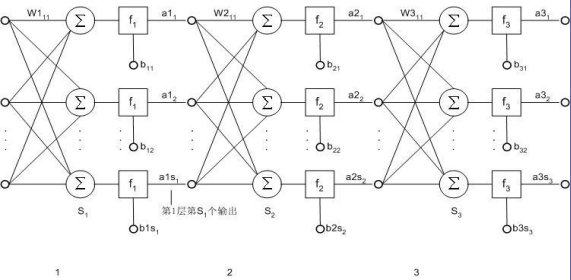
此时的输出矢量为一个维数为 $S \times q$

的矩阵 $A_{s \times q}$



多层神经网络

- 将两个以上的单层神经网络联接起来则组成多层神经网络。三层的神经网络结构图。



多层神经网络

- r个输入矢量
- S_1 第一层神经元个数
- S_2 第二层神经元个数
- S_3 第三层神经元个数

一般情况下，不同层有不同的神经元数
每个神经元有一个为常数1的偏差值
最后一层为输出层
中间为隐含层

在多层网络中，每一隐含层的输出都是下一层的输入



BP神经网络

- BP网络 (Back-Propagation Network, 反向传播)

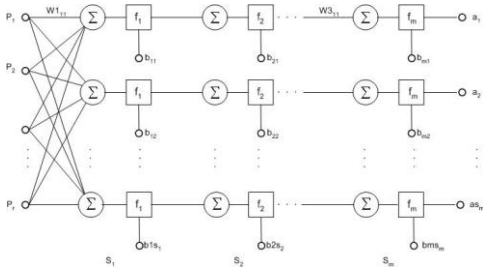
- BP算法: 包括信息正向传播，误差反向传播两部分；
 - 信息正向传播: 输入样本由输入层经中间层向输出层的计算过程，此阶段是连接权值固定，逐层计算各节点的输出，当输出层的实际输出与希望输出相差较大，则开始下面第二过程。
 - 误差反向传播: 在此阶段，各节点的输出保持不变，从输出层开始反向逐层逐节点计算各连接权值的修改量，直到输入层为止。
 - BP网络权值修正通过以上二个反复交替的学习过程，最终使输出层的实际输出与希望输出趋于一致。
- 在人工神经网络的实际应用中，80%~90%的人工神经网络是采用BP网络或它的变化形式。



BP神经网络：BP网络的结构

● BP网络的结构

- 单向传播的多层前馈网络
- 由输入层、输出层和若干个隐含层组成（至少一个）
- 同层节点间无任何耦合连接



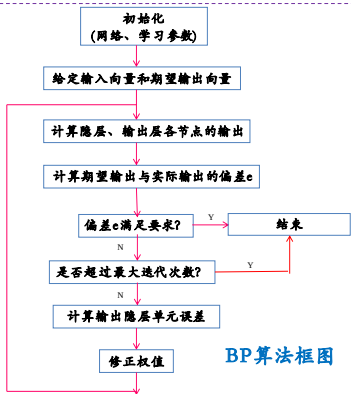
BP神经网络：BP算法

● BP算法的主要思想

- 1) 对于 r 个输入学习样本： $P_1, P_2, \dots, P_r$,
- 2) 已知与其对应的输出样本为： $T_1, T_2, \dots, T_r$ 。
- 3) 学习的目的是用网络的实际输出 $A_1, A_2, \dots, A_r$ 与目标矢量 $T_1, T_2, \dots, T_r$ 之间的误差来修改其权值，使 A_i ($i=1, 2, \dots, r$) 与期望的 T_i 尽可能接近，即：**网络输出层的误差平方和达到最小。**
- 4) 通过连续不断地在相对于误差函数斜率下降的方向上计算网络权值和偏差的变化而逐渐逼近目标；
- 5) 每一次数值和偏差的变化都与网络误差的影响成正比，并以反向传播的方式传递到每一层。



BP神经网络：BP算法



BP神经网络：BP网络设计

● BP网络的设计

- 在设计BP网络前，应了解ANN适应范围
 - 传统的计算技术不适合。
 - 所求解的问题要求是定性的，或是复杂的映射，没有精确数量表示的高级相互依赖关系。
 - 数据有很多噪音或错误。
- ANN设计的基本思路
 - 根据应用要求，选取样本集，确定ANN模型结构。
 - 用样本集去训练、测试所设计的网络。
 - **关键点：结构、样本和训练设计。**



BP神经网络：BP网络设计

- BP网络的结构设计

● 输入层和输出层的设计

- ✓ **输入层设计：**它的神经元数是依所求的问题和数据表示方式而定，输入数据有不同形式（图形，声音……模拟量、数字量等）但ANN只能处理数字输入数据）。
- ✓ **输出层设计：**ANN输出有是非判断、分类、模式识别、实数与类型。

● 传递函数

- ✓ 一般选S形函数

● 隐层数目的设计

- ✓ 一般采用一个隐层，有时可考虑采用双隐层，不宜采用二层以上的隐层。



BP神经网络：BP网络设计

- 隐层单元数的确定

- 关系网络成败的关键问题，其确定十分复杂，目前尚无现成的解析式表示，与问题的要求，与输入、出单元多少等有关。
- **原则：**在能正确反映输入/输出关系基础上，尽管选取较少的隐层单元数。
 - ✓ 隐层单元数太少，难于处理复杂问题；太多，训练时间长可能产生训练过度
 - ✓ 确定隐层单元数经验公式，例如：
 - $h = \text{int}[(\text{输入单元数} + \text{输出单元数})/2 + a]$ (其中 a 为0~10之间)
 - $h = \text{sqrt}(\text{输出单元数} * \text{输入单元数})$
 - ✓ 更多试验



BP神经网络：BP网络设计

网络初始权值的确定

- 由于BP网络是非线性，故初始权值对是否收敛关系很大，初始权值的采用经验选取。
- 由一定算法确定或取随机数。
- 利用应用软件MATLAB的神经网络工具箱提供的隐层节点的权值初始化函数产生，效果好。



BP神经网络：学习样本的设计

学习（训练）样本的设计

- **学习样本数据必须满足两个条件**
 - 样本数据组中必须包括全部模式
 - 要将可能出现的随机噪声考虑进去。
- **学习样本数的确定**
 - 样本数越大，一般越能正确反映输入输出关系。但太多增加了收集、分析的代价。
 - 样本数主要取决于网络大小，输入输出的分布及网络测试的需要。
 - **经验规则**：训练样本数为网络连接权总数的5~10倍。
- **网络学习前学习样本数据的处理**
 - **收集有关数据**：深入了解问题，收集可能的有关输入样本
 - **评价和提取有效变量**：通过评价，提取解决问题最有效的变量，以建立一个简单有效的网络模型。
 - **样本数据变换处理**：一般而言，样本为正态分布时，学习最有效，若样本不是正态分布，可进行非线性变换；进行尺度变换：进行归一化处理。



BP神经网络：学习样本的设计

神经网络的训练（学习）与检验

- 将样本数据分为二部分，2/3用于训练，1/3用于检验。
- **训练**：仅将输入为输入网络、计算网络输出，与输入的希望输出比较，依误差调整，直至误差小于一定值时，为所确定的网络。
- **检验**：利用已训练的网络，保持不变，用检验样本输入，正向运行该网络，检验输出的均方误差。



7.5 MATLAB神经网络工具箱

Matlab NN Toolbox (神经网络工具箱)

- 提供了很多经典的学习算法，能够快速实现对实际问题的建模求解。
- 涵盖的神经网络模型：
 - 感知器网络
 - 线性神经网络
 - BP网络
 - 径向基函数网络
 - 反馈网络
 - 自组织网络
 - 控制系统网络模型



1) BP网络初始化

■ 最多三层

● 单层BP神经网络

[w,b]=initff(p,s,f)

- w: 权重阵
- b: 偏差矩阵
- p: 输入矢量
- s: 神经元个数
- f: 激活函数



1) BP网络初始化

● 二层BP神经网络

[w1,b1,w2,b2]=initff(p, s1, f1, s2, f2)

- s1: 第一层神经元个数
- s2: 第二层神经元个数

● 三层BP神经网络

[w1,b1,w2,b2,w3,b3]=initff(p, s1, f1, s2, f2, s3, f3)

例:

P=[-2 2; 0 5]

[w1,b1,w2,b2]=initff(P,2,'tansig',3,'purelin')

运行结果得到w1,b1,w2,b2打印结果

tansig: 双曲正切S形

purelin: 纯线性 (NetInff.m)



1) BP网络初始化

```
w1 =  
    0.2373    0.3475  
   -0.4948    0.0092  
  
b1 =  
   -0.4540  
   -0.1639  
  
w2 =  
   -0.3908    0.3644  
   -0.6207   -0.3945  
   -0.6131    0.0833  
  
b2 =  
   -0.6983  
    0.3958  
   -0.2433
```



2) 激活函数

```
1) P=[0.1 0.4 -1.2];  
   a=purelin(p)      线性传递函数  
   (NetPurelin.m)  
结果  
   a=[0.1 0.4 -1.2]  
  
2) P=[-1.0 -0.5 0.0;  
      1.0 2.0 3.0];  
   a=hardlim(p)      硬限传递函数  
   (NetHardlim.m)  
   w*p+b>=0  A=1  
   w*p+b<0   A=0  
结果  
   a =  
       0       0       1  
       1       1       1 ]
```



2) 激活函数

```
3) P=[-1.0 2.1 0.3 0.0];  
   a=logsig(P)      对数S型传递函数  
                   a为0至1之间的值  
   (NetLogsig.m)  
结果: a=0.2689  
       0.8909  
       0.5744  
       0.5000  
  
4) P=[-2.1; -1.0; 0.2; 1.0; 2.0];  
   a=tansig(P)      双曲正切传递函数  
                   a为-1至1之间的值  
   (NetTansig.m)  
结果: a=- 0.9705  
       -0.7616  
       0.1974  
       0.7616  
       0.9640
```



3) 训练函数

```
trainbp  利用BP算法训练前向网络  
trainbpx 利用快速BP算法训练前向网络  
  
[w, b, te, tr]=trainbpx(w, b, 'f', p, t, tp)  一层BP网络  
  
[w1, b1, w2, b2, te, tr]=trainbpx(w1, b1, 'f1', w2, b2, 'f2', p, t, tp)  
                                           二层BP网络  
  
[w1, b1, w2, b2, w3, b3, te, tr]  
   =trainbpx(w1, b1, 'f1', w2, b2, 'f2', w3, b3, 'f3', p, t, tp)  
                                           三层BP网络
```



3) 训练函数

tp 训练参数 (train parameter) 为8个数

tp(1) 指定两次显示间的训练次数, 默认值为25次

tp(2) 指定训练的最大次数, 默认值为100次

tp(3) 指定误差平方和指标, 默认值为0.02

tp(4) 指定学习速率, 默认值为0.01

tp(5) 指定学习速率增加的比率, 默认值为1.05

tp(6) 指定学习速率减少的比率, 默认值为0.7

tp(7) 指定动量常数, 默认值为0.9

BP网络的改进: 附加动量法。允许网络忽略网络上的微小变化特性, 以便滑过这些局部极小值。默认值为0.4。

tp(8) 指定最大误差比率



3) 训练函数

• NaN代表取默认值

例 tp=[disp-freq max-epoch err-goal NaN
 NaN NaN NaN err-ratio]

一旦训练达到了最大训练次数, 或网络误差平方和小于期望误差, 即使会使网络停止学习。

左边为训练后的权值矩阵W, 阈值矢量b。

te: 实际训练步数。

tr: 第一行: 训练过程中网络的误差。
 第二行: 相应的自适应学习速率。

3) 训练函数

例 训练一个网络 NetTrain.m

```
p = [-2 2; 0 5]; % 2组2个输入
t = [-3 0 3; 0 5 9]; % 2组3个输出

[w1,b1,w2,b2] = initff(p,4,'tansig',3,'purelin','') %初始化
disp-freq=1; %两次训练间显示次数
max-epoch=100; %训练最大次数
err-goal=0.001; %误差平方和指标
lr=1; %学习速率
momentum=0.95 %动量常数
tp=[disp-freq max-epoch err-goal lr...
NaN NaN momentum NaN];
[w1,b1,w2,b2,te,tf]=trainbpz(w1,b1,'logsig',w2,b2,'purelin',p,t,tp)
```

4) 模拟一个网络 (NetSimu.m)

在上例基础上:

```
a=simuff(p,w1,b1,'tansig',w2,b2,'purelin')
```

结果 a = -4.4615 1.6511
-7.9840 2.1060
-8.8829 3.2378

5) 竞争函数 (NetCompet.m)

compet(A)返回一个输出矢量矩阵，每一列矢量中仅包含一个1。位于对应的响应矢量在网络输入矢量矩阵N中有最大值的位置，而其余元素为0

```
P=[0.0; 0.2; 0.6; 0.1];
a=compet(P) (Competition)
```

结果 a=(3,1) 1

第3行第1列元素为1,该列中最大值取为1，其余值取为0

```
a=0
0
1
0
```

5) 竞争函数 (NetCompet.m)

%例题(16组，5个输入)

```
p1=[48.5330 101.9800 137.0600 171.4400 103.8300;
34.1180 66.5560 86.8130 103.7100 62.6370;
27.7700 51.5670 66.0520 76.2980 40.8070;
32.9570 67.3050 86.9300 97.3910 46.9050;
36.3420 88.3400 121.8300 145.7900 72.1920;
25.5830 52.5500 67.7960 82.5640 50.2550;
25.7270 61.7980 85.6250 116.7900 77.0550;
30.2370 64.8760 86.1020 104.0100 57.6620;
22.8070 49.0750 64.2290 75.3760 40.5380;
29.2200 53.3010 68.1530 82.8420 50.4550;
42.8360 98.9110 135.0900 170.9200 98.6860;
36.1610 92.1400 129.0700 167.3300 97.5290;
39.0750 94.3740 131.1200 168.7400 98.0740;
27.3600 63.7740 87.3270 117.6400 77.2910;
32.0600 64.0050 84.0710 96.2870 46.7920;
29.6070 65.2520 88.5780 118.4500 77.6270;]
```

5) 竞争函数 (NetCompet.m)

(16组，4个输出)

```
t1=[ 1, 1, 1, 2, 2, 3, 3, 2, 3, 1, 2, 3, 2, 3, 1, 2;
1, 2, 3, 1, 3, 1, 3, 2, 3, 2, 1, 2, 2, 2, 3, 2;
1, 2, 3, 2, 1, 3, 2, 2, 3, 3, 1, 1, 1, 2, 2, 2;
1, 2, 3, 3, 2, 2, 1, 2, 3, 2, 1, 1, 1, 1, 3, 1];
```

```
pp=[ 36.9;
73.0;
102.7;
132.6;
56.5];
```

5) 竞争函数 (NetCompet.m)

```
[W1,b1,W2,b2] = initff(p1,10,'tansig',t1,'purelin');
```

% t1-取目标向量中的行数(4)，即输出个数作为输出
神经元个数

```
W2 = W2*0.01;
b2 = b2*0.01;
pause
```

```
df = 20;
mc = 5000;
eg = 0.1;
mc = 0.95;
tp = [df mc eg NaN NaN NaN mc];
```




MATLAB的神经网络应用实例

2) 网络结构

输入：用5×7个数字（0或1）表示一个英文字母，
∴输入节点为35个。

输出层：26个神经元

隐含层：取1层，用10个神经元。

即：**35-10-26结构**。

（理论上对隐含层未明确规定。一般要试算，对不同神经元数进行对比，如取35, 26, 20, 10）。



MATLAB的神经网络应用实例

辨识训练精度 **无噪声err-goal=0.1**

训练次数 **5000次**

激活函数 **输入在（0，1）范围的**

对数S型激活函数logsig。

噪声问题：

噪声（污点）可使网络的输出值0或1不正确。

用带噪声网络训练

将输出经过一层**竞争网络函数Compet**的处理，使网络的输出只在最接近输入值的输出位置输出为1。保证在其他位置输出均为0。



MATLAB的神经网络应用实例

源程序：

```
clf; %清除当前图
figure(gcf) %打开或创建图形窗口(gcf返回当前图形窗口的句柄)
echo on %打开回显开关
pause %暂停、等待输入任意键
LetterA=[0 0 1 0 0...
0 1 0 1 0...
0 1 0 1 0
1 0 0 0 1...
1 1 1 1 1...
1 0 0 0 1...
1 0 0 0 1]';
.....
```



MATLAB的神经网络应用实例

```
LetterZ=[1 1 1 1 1...
0 0 0 0 1...
0 0 0 1 0...
0 0 1 0 0...
0 1 0 0 0...
1 0 0 0 0...
1 1 1 1 1];
alphabet=[letterA, letterB, ... letterZ]; %输入向量矩阵 26
组（列） 35个（行）
targets=eye(26); %目标矩阵为26×26对角阵
```

即 $targets = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}_{26 \times 26}$



MATLAB的神经网络应用实例

```
[R,Q]=size(alphabet); %求矩阵alphabet的行数列数R（R：输入向量个数，Q：输入向量组数）
```

pause

```
s1=10; %隐含层神经元取为10个
```

```
[w1,b1,w2,b2]=initff(alphabet,s1,'logsig',targets,'logsig');
```

%网络初始化： alphabet 输入向量

s1 隐含层神经元个数

logsig 激活函数

targets 目标函数

w1 隐含层的权值矩阵

b1 隐含层偏差矩阵

w2 隐含层输出到输出层的输入向量权矩阵

b2 输出层的偏差矩阵



MATLAB的神经网络应用实例

```
w2=w2*0.01; %对w2,b2再做调整
```

```
b2=b2*0.01;
```

Pause

```
df=20; %训练过程的显示频率（df=disp-freq; 每20步显示一次）
```

```
me=5000; %最多训练5000步（me=max-epoch）
```

```
eg=0.1; %误差指标（eg=err-goal）
```

```
mc=0.95; %动量因子
```

（BP网络的改进，附加动量法忽略网络上的微小变化特征，以便滑过这些局部极值点）



MATLAB的神经网络应用实例

做法：在每一个权值的变化上加上一项正比于前次权值的变化量的值，并根据反向传播法来产生新的权值变化。

$$\Delta w_{ij}(k+1) = (1 - mc)\eta\delta_i P_j + mc\Delta w_{ij}(k)$$

$$\Delta b_i(k+1) = (1 - mc)\eta\delta_i + mc\Delta b_i(k)$$

式中k为训练次数，mc为动量因子，一般取0.95左右。

```
tp=[df me eg NaN NaN NaN mc]; %训练开始
P=alphabet; %输入矢量
T=targets; %目标矢量
```



MATLAB的神经网络应用实例

(理想输入—第一个网络)

```
[w1,b1,w2,b2,ep, tr]=trainbpx(w1,b1,'logsig',
                               w2,b2,'logsig',P,T,tp);
```

(用快速BP算法训练网络)

(ep 网络实际迭代次数;

tr 矩阵，第一行记录迭代过程中的误差;

第二行记录迭代过程自适应学习率)

pause

%以下开始采用有噪声样本对网络训练

w1n=w1;

%W1n表示有噪声Noise训练的有关参数



MATLAB的神经网络应用实例

```
b1n=b1;
w2n=w2;
b2n=b2;
me=300; %最多训练300步
eg=0.6; %误差指标
tp=[df me eg]; %除df,me,eg外的其余值，取缺省值。
pause %开始训练
T=[targets targets targets targets];
(两组理想输入 两组带噪音输入四组目标矩阵)
```

$$T = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & & 1 & & 1 & & 1 & \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

4组 26行 104列



MATLAB的神经网络应用实例

```
for pass=1:10 (10组)
    fprintf('Pass=%d of %d\n',pass);
    P=[alphabet,alphabet,(alphabet+randn(R,Q)*0.1),...
      (alphabet+randn(R,Q)*0.2)];
    (正态分布随机阵，两组带噪音。)
```

```
[W1n,bln,W2n,b2n,ep,tr]=trainbpx(W1n,bln,
                                   'logsig',W2n,b2n,'logsig',P,T,tp);
```

end

$$P = \begin{bmatrix} \text{35行104列} \\ \text{alphabet} & \text{alphabet} & \text{alphabet} + (\dots) & \text{alphabet}(\dots) \\ \text{2组无噪音} & & \text{2组有噪音} & \end{bmatrix}$$

保持网络同时对理想输入分类的能力



MATLAB的神经网络应用实例

```
pause %以下再次采用无噪声样本
      对第二个网络进行训练
df=5; %训练过程的显示频率
me=500; %最多训练500步
eg=0.1; %误差指标
tp=[df me eg];
P=alphabet;
T=targets;
[W1n,b1n,W2n,b2n,ep, tr]
=trainbpx(W1n,bln,'logsig',
          W2n,b2n,'logsig',P,T,tp);
```



MATLAB的神经网络应用实例

```
P1=[0000.....1000000000;
     0000.....1000000000;
     .....
     0000.....1000000000]
(26行35列)
```

```
P2=[001001.....110001; A
     001001.....110001; A
     111101.....111110; B
     001001.....110001; A
     111101.....111110; B
     .....],
P=P2+P1;
```



MATLAB的神经网络应用实例

```
(T=targets; )
A=simuff (P,w1n,b1n,'logsig',w2n,b2n,'logsig') ;
%模拟一个BP网络，A中为输出值
AA=compet (A) ;
```



MATLAB的神经网络应用实例

```
pause %以下检测网络性能
noise-range=0:0.05:0.5; (10组)
max-test=100;
network1=[];
network2=[];
T=targets; %以下开始检测
for noiselevel=noise-range (0-0.5共10组)
fprintf('Testing networks with noise level of %.2f.\n',
noiselevel);
error1=0;
error2=0;
```



MATLAB的神经网络应用实例

```
for i=1:max-test (随机产生100个输入矢量)
P=alphabet+randn(35,26)*noiselevel;
%以下测试第一个网络
A=simuff(P,W1,b1,'logsig',W2,b2,'logsig');
%无噪音（模拟一个前向网络，返回网络层的输出A）
AA=compet (A) ;
errors1=errors1+sum(sum(abs(AA-T)))/2;
%测试第二个网络
An=simuff(P,W1n,b1n,'logsig',W2n,b2n,'logsig');
AAn=compet(An);
errors2=errors2+sum(sum(abs(AAn-T)))/2;
End
(sum(a):求各列元素的总和)
```



MATLAB的神经网络应用实例

```
network1=[network1 error1/26/100];
network2=[network2 error2/26/100];
End
pause %显示测试结果
clf
plot(noise-range,network1*100,'-l',
noise-range,network2*100);
title('percentage of Recognition Errors');
xlabel('Noise level');
ylabel('Network1--Network2');
echo off
```



小结

(1) 为训练一个网络，使其具有抗干扰能力，

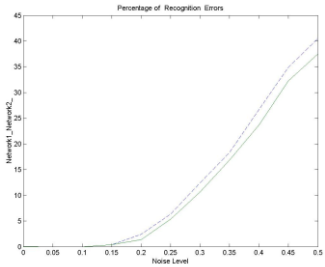
具体做法：

- 先训练一个具有较低误差平方和的具有理想输入的神经网络。
- 用10组“理想+随机噪声”的输入矢量训练网络，每一次训练后的权矢量作为下一组输入矢量训练的初始值。
- 防止因抗干扰性而牺牲了对理想输入识别的正确性。
∴以训练后的权矢量作为初始权值，对无噪声输入再进行训练。



小结

(2) 测试结果



- 平均值为0-0.1之间的噪音影响，能几乎100%地识别。
- 噪音较大时如0.1~0.2，理想网络（无噪音--）比带噪声训练的出错率高



小结

(3) 进一步改进:

- a.把 5×7 点阵改为 10×14 点阵
- b.使训练精度更高,或增加隐含层中的神经元数

结论:

网络设计的好坏,取决于设计者对问题的认识,设计思想及运用MATLAB工具箱的能力。



BP网络设计分析

好坏标准: 精度、训练时间

(1) 网络的层数

理论上已经证明: 具有偏差和至少一个S型隐含值加上一个线性输出层的网络, 能够逼近任何有理函数。
增加层数可以进一步降低误差, 但同时使网络复杂化, 增加权值训练时间。

增加隐含层的神经元个数也可以降低误差, 其训练效果比增加层数更易观察和调整。

∴ 宜优先考虑增加隐含层的神经元个数。



BP网络设计分析

(2) 隐含层的神经元个数

一般地, 隐含层的神经元个数越多、功能越大, 但所需训练时间也随之增加。

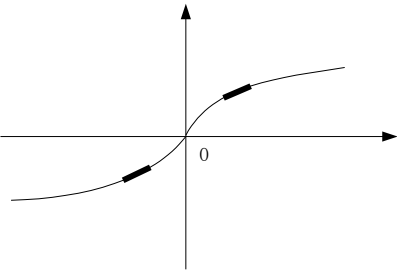
• 选择原则: 在能够解决问题的前提下, 再加上1至2个神经元以加快误差的下降。

(3) 初始权值的选取

初始权值对能否收敛及训练时间的长短影响很大。一般在激活函数变化最大处调节。



BP网络设计分析



在MATLAB工具箱中初始化权值W1和B1的函数。



BP网络设计分析

(4) 学习速率:

学习速率决定每一次循环训练中所产生的权值变化量。

- 学习速率大: 系统不稳定
- 学习速率小: 收敛慢、训练时间长
- 一般地, 取0.01~0.8。

(5) 学习样本的顺序:

实验结果表明, BP网络接受样本的顺序对训练结果有较大影响。对于与该样本序列较后的样本较接近的输入, 网络的输出精度明显高于与样本序列较前的样本较接近的输入对应的输出精度。

(排在较前的样本对网络的影响会被排在较后的样本的影响掩盖掉。)



BP网络设计分析

(6) BP网络的精度:

将学习样本的2/3作为学习样本来训练网络;

留取学习样本的1/3作为输入来校验网络的正确性。



本章学习重点

- 了解智能系统的概念、特征和基本原理
- 了解知识的表示和处理
- 了解知识库系统的概念
- 了解神经网络的概念和作用
- 掌握使用MATLAB进行BP神经网络的应用
- 能使用神经网络解决土木建筑领域中的现实问题



提高内容参考

- 自己开发一个BP神经网络程序，包括样本训练、误差控制及结果输出；
- 用MATLAB神经网络工具箱中的BP神经网络解决一个土木工程中的实际问题；
- 开发一个知识库，用于指导建筑物种机电设备的安装和运营；
- 研究建筑施工中的语义识别技术，提出有效知识的提取算法。



谢谢！

清华大学土木工程系

胡振中

huzhenzhong@tsinghua.edu.cn