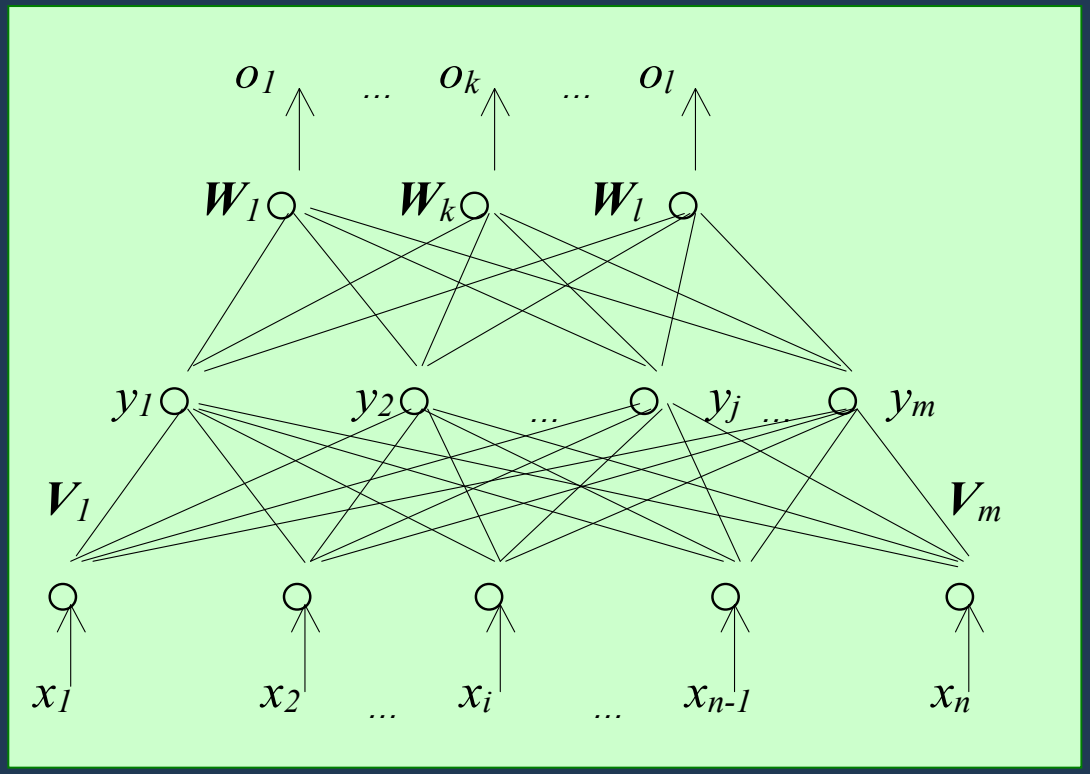
第三章 前馈人工神经网络

——误差反传（BP）算法的改进与BP网络设计

3.4 基于 BP 算法的多层前馈网络模型

三层 BP 网络



模型的数学表达

输入向量： $X=(x_1,x_2,\ldots,x_i,\ldots,x_n)^T$

隐层输出向量： $Y=(y_1,y_2,\ldots,y_j,\ldots,y_m)^T$

输出层输出向量： $O=(o_1,o_2,\ldots,o_k,\ldots,o_l)^T$

期望输出向量： $d=(d_1,\ d_2,\ldots,d_k,\ldots,d_l)^T$

输入层到隐层之间的权值矩阵： $V=(V_1,V_2,\ldots,V_j,\ldots,V_m)$

隐层到输出层之间的权值矩阵： $W=(W_1,W_2,\ldots,W_k,\ldots,W_l)$

各个变量之间如何建立联系，来描述整个网络？

神经网络的学习

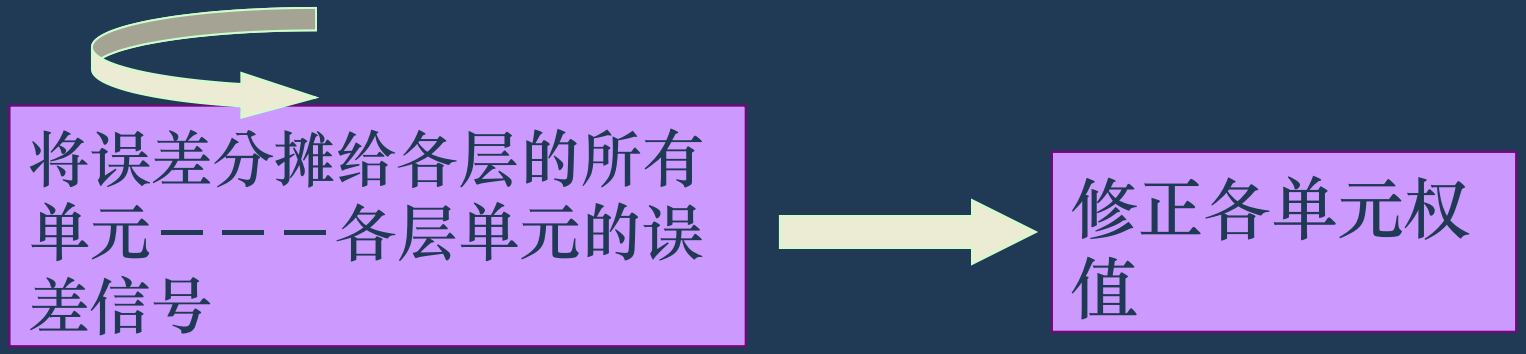
- 学习的过程：
  - 神经网络在外界输入样本的刺激下不断改变网络的连接权值乃至拓扑结构，以使网络的输出不断地接近期望的输出。
- 学习的本质：
  - 对可变权值的动态调整
- 学习规则：

$$\Delta W_j = \eta r[W_j(t), X(t), d_j(t)]X(t)$$

  - 权值调整规则，即在学习过程中网络中各神经元的连接权变化所依据的一定的调整规则。
- BP 算法是一种学习规则

BP 算法的基本思想

- 学习的类型：有导师学习
- 核心思想：
  - 将输出误差*以某种形式*通过隐层向输入层逐层反传



- 学习的过程：
  - 信号的正向传播　　误差的反向传播

BP 算法的学习过程

- 正向传播：
  - 输入样本 - - - 输入层 - - - 各隐层 - - - 输出层
- 判断是否转入反向传播阶段：
  - 若输出层的实际输出与期望的输出（教师信号）不符
- 误差反传
  - 误差以某种形式在各层表示 - - - - 修正各层单元的权值

- 网络输出的误差减少到可接受的程度
  进行到预先设定的学习次数为止

建立权值变化量与误差之间的关系

□输出层与隐层之间的连接权值调整

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$$

$$j=0,1,2,\ldots,m;\ \ k=1,2,\ldots,l \qquad (3.4.9a)$$

□隐层和输入层之间的连接权值调整

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}}$$

$$i=0,1,2,\ldots,n;\ \ j=1,2,\ldots,m \qquad (3.4.9b)$$

式中负号表示梯度下降，常数 $\eta \in (0,1)$ 表示比例系数，反映了训练速率。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 (Gradient Descent) 算法。

BP 算法的程序实现

(1) 初始化；

(2) 输入训练样本对 $X \Leftarrow X^p$ 、 $d \Leftarrow d^p$

计算各层输出；

(3) 计算网络输出误差；

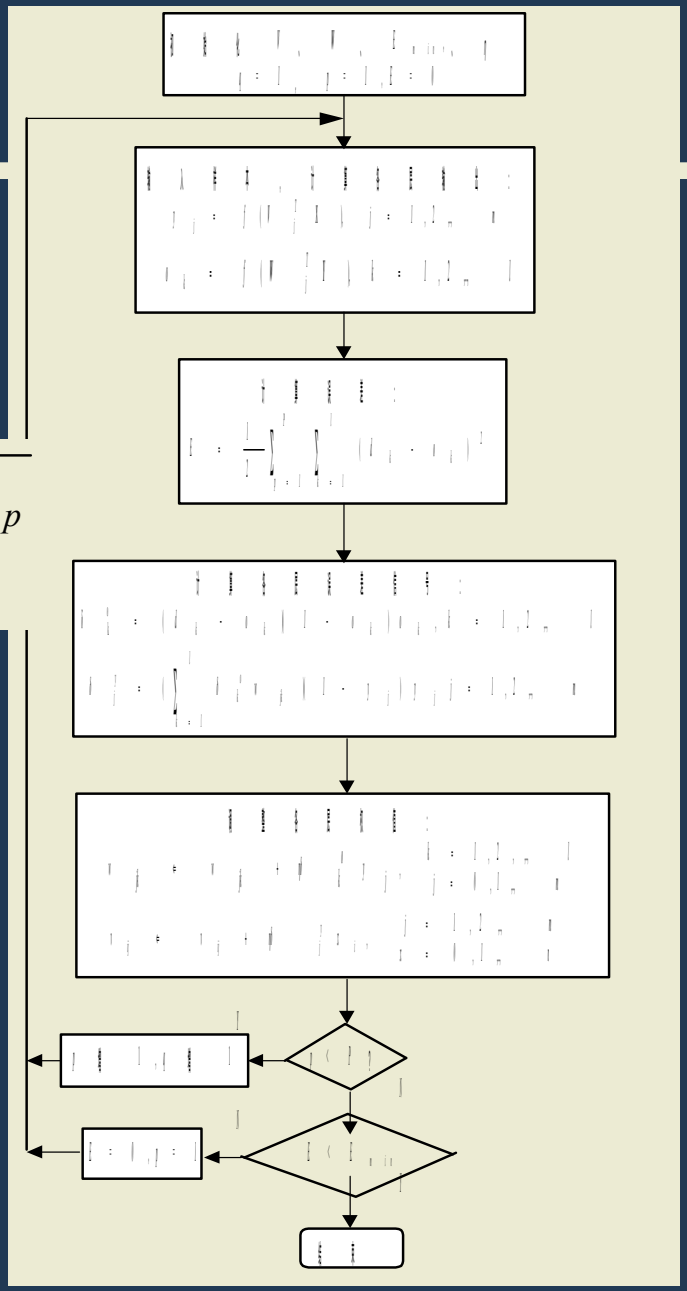
$$E_{RME} = \sqrt{\frac{1}{P}\sum_{p=1}^{P} E^p}$$

(4) 计算各层误差信号；

(5) 调整各层权值；

(6) 检查是否对所有样本完成一次轮训；

(7) 检查网络总误差是否达到精度要求。

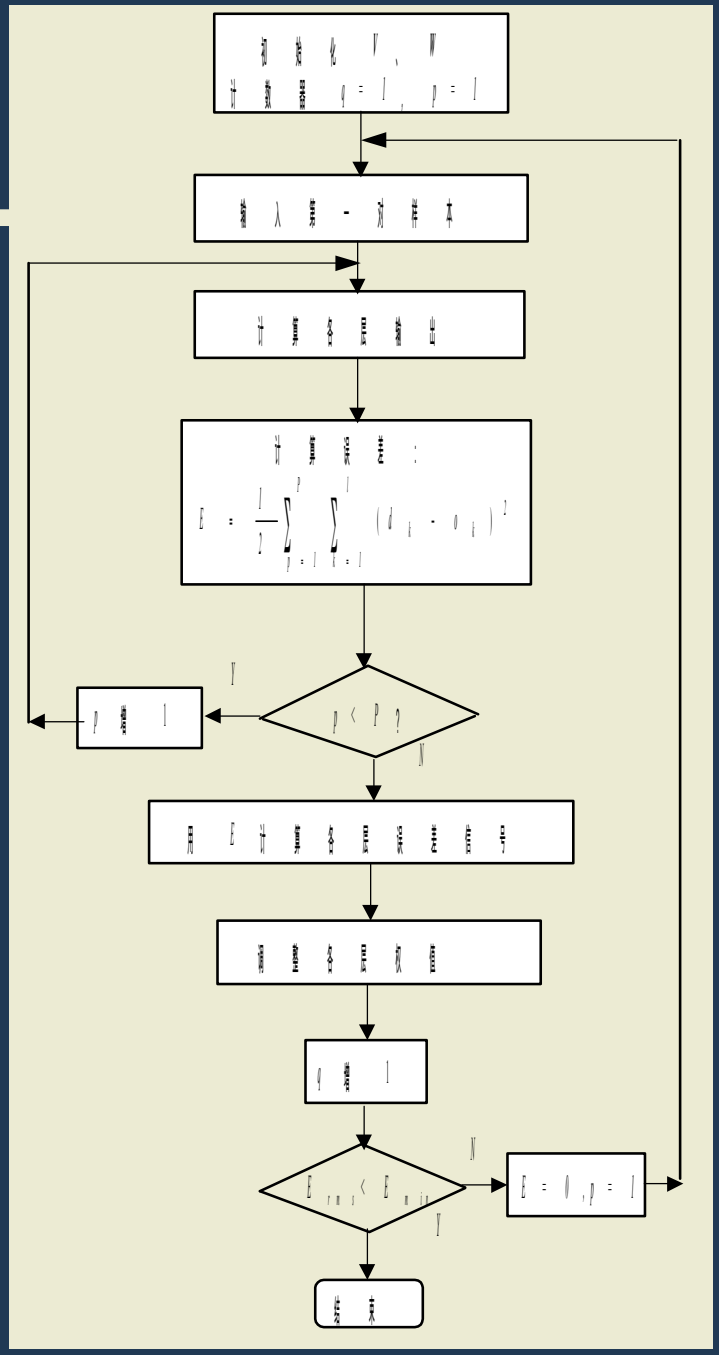


BP 算法的程序实现

另一种方法是在所有样本输入之后，计算网络的总误差：

$$E_{总}=\frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{l}(d_k^p-o_k^p)^2$$

然后根据总误差计算各层的误差信号并调整权值。



多层前馈网的主要能力

(1) 非线性映射能力

多层前馈网能学习和存贮大量输入 - 输出模式映射关系，而无需事先了解描述这种映射关系的数学方程。只要能提供足够多的样本模式对供 BP 网络进行学习训练，它便能完成由 n 维输入空间到 m 维输出空间的非线性映射。

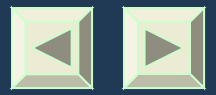
多层前馈网的主要能力

(2) 泛化能力

当向网络输入训练时未曾见过的非样本数据时，网络也能完成由输入空间向输出空间的正确映射。这种能力称为多层前馈网的泛化能力。

(3) 容错能力

输入样本中带有较大的误差甚至个别错误对网络的输入输出规律影响很小。

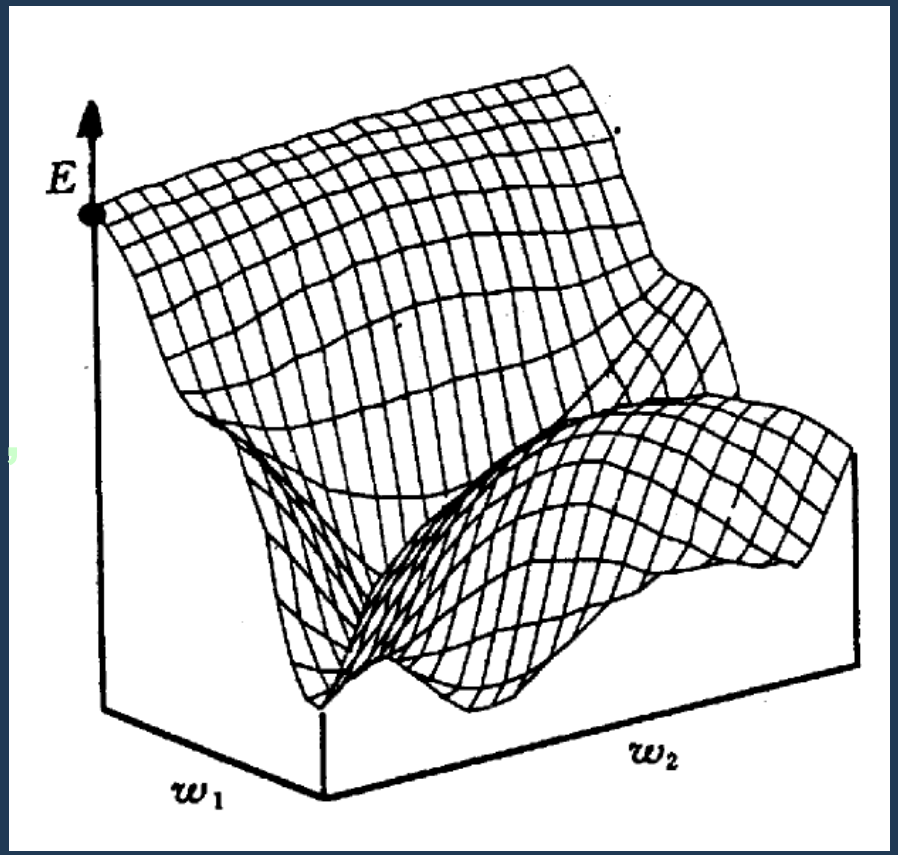


误差曲面与 BP 算法的局限性

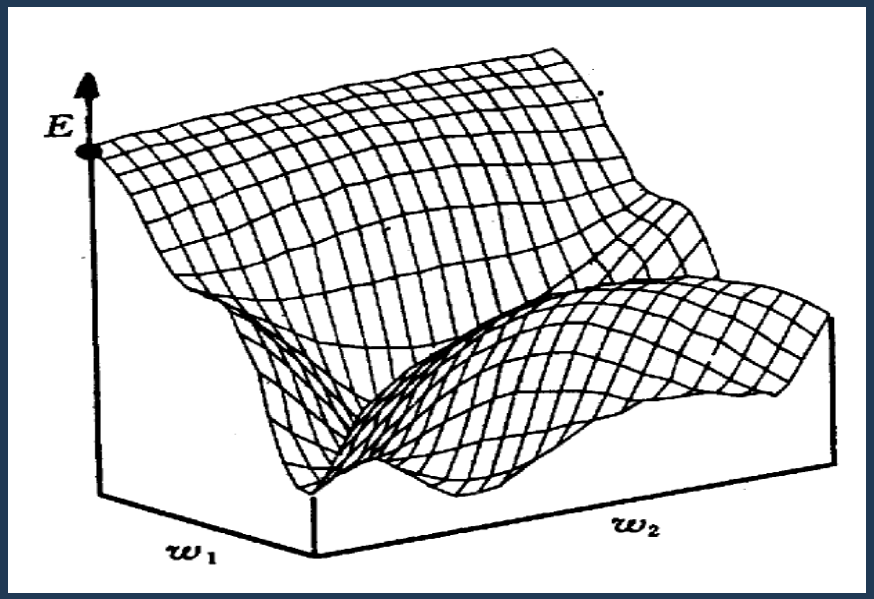
误差函数的可调整参数的个数 n_w 等于各层权值数加上阈值数，即：

$$n_w = m \times (n+1) + l \times (m+1)$$

误差 E 是 n_w+1 维空间中一个形状极为复杂的曲面，该曲面上的每个点的“高度”对应于一个误差值，每个点的坐标向量对应着 n_w 个权值，因此称这样的空间为误差的权空间。



误差曲面的分布——BP 算法的局限性

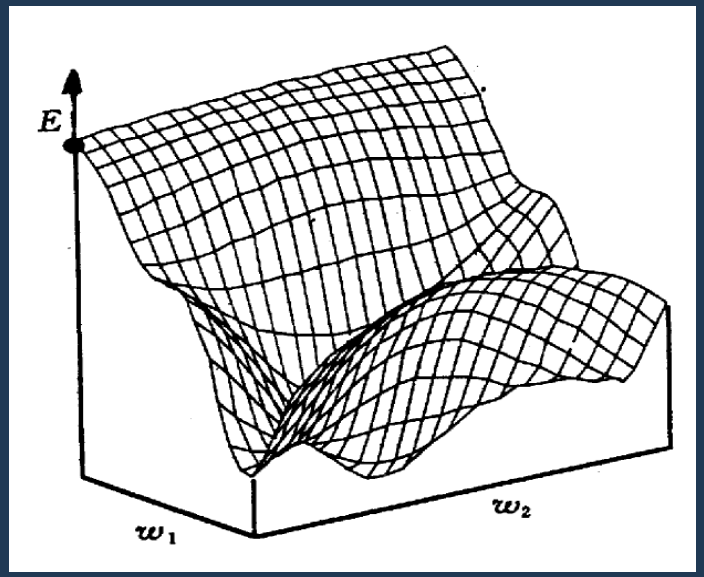


- 曲面的分布特点 -------- 算法的局限性
- (1) 存在平坦区域 -------- 误差下降缓慢，影响收敛速度
- (2) 存在多个极小点 ------ 易陷入局部最小点

曲面分布特点1：存在平坦区域

- 平坦 - - 误差的梯度变化小 - - δ_k^o 接近于零

$$\frac{\partial E}{\partial w_{ik}} = -\delta_k^o y_j$$



存在平坦区域的原因分析

- δ_k^o 接近于零的情况分析

$$\delta_k^o = (d_k - o_k)o_k(1 - o_k)$$

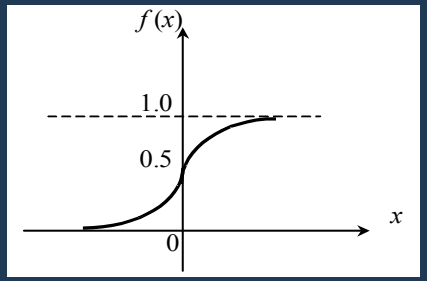
一种可能是 o_k 充分接近 d_k

第二种可能是 o_k 始终接近 0

第三种可能是 o_k 始终接近 1

对应着误差的某个谷点

平坦区

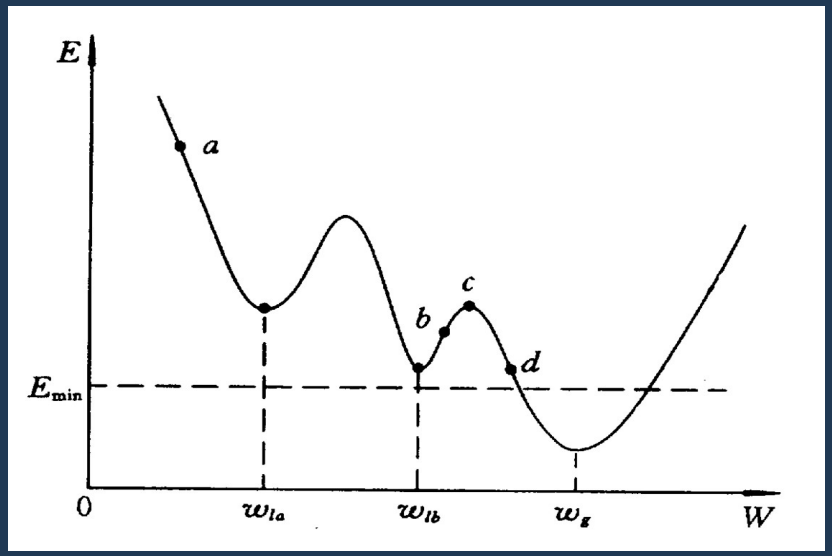


- 造成平坦区的原因：

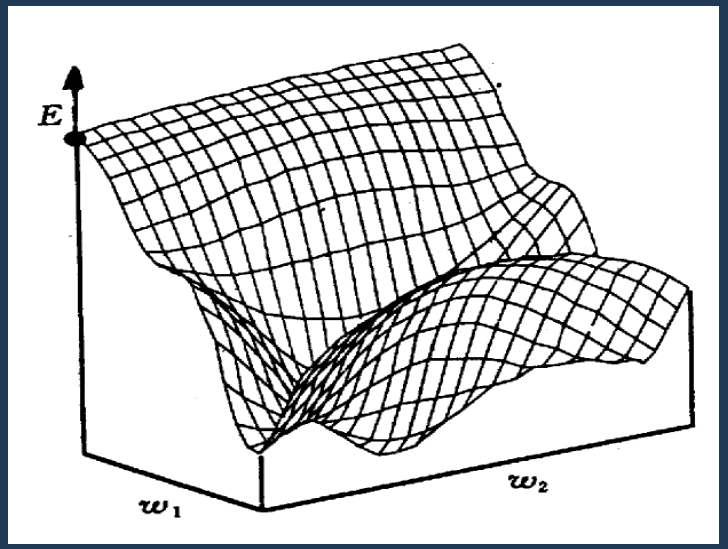
各节点的净输入过大

$$\left| \sum_{j=0}^{m} w_{jk} y_j \right| > 3$$

曲面分布特点2：存在多个极小点



单权值



双权值

- 误差梯度为零
- 多数极小点都是局部极小，即使是全局极小往往也不是唯一的。

曲面分布特点2：存在多个极小点

- BP 算法
  - - - 以误差梯度下降为权值调整原则
- 误差曲面的这一特点
  - - - 使之无法辨别极小点的性质
- 导致的结果：
  - 因而训练经常陷入某个局部极小点而不能自拔，从而使训练无法收敛于给定误差。

标准 BP 算法的改进——引言

- 误差曲面的形状 - - 固有的
- 算法的作用是什么？
  - 调整权值，找到最优点
- 那么如何更好地调整权值？
  - 利用算法使得权值在更新的过程中，‘走’合适的路径，比如跳出平坦区来提高收敛速度，跳出局部最小点等等
- 如何操作？
  - 需要在进入平坦区或局部最小点时进行一些判断，通过改变某些参数来使得权值的调整更为合理。

标准的 BP 算法内在的缺陷：

⑴ 易形成局部极小而得不到全局最优；

⑵ 训练次数多使得学习效率低，收敛速度慢；

⑶ 隐节点的选取缺乏理论指导；

⑷ 训练时学习新样本有遗忘旧样本的趋势。

针对上述问题，国内外已提出不少有效的改进算法，下面仅介绍其中 3 种较常用的方法。

3.5 标准 BP 算法的改进

- 改进1 ：增加动量项
- 改进2：自适应调节学习率
- 改进3： 引入陡度因子

改进1 ：增加动量项

- 提出的原因：
  - 标准 BP 算法只按 t 时刻误差的梯度降方向调整，而没有考虑 t 时刻以前的梯度方向
  - - - - 从而常使训练过程发生振荡，收敛缓慢。
- 方法：

$$\Delta \mathbf{W}(t) = \eta \delta \ \mathbf{X} + \alpha \Delta \mathbf{W}(t-1)$$

α 为动量系数，一般有 $\alpha \in (0, 1)$

改进1 ：增加动量项

- 实质：
  - 从前一次权值调整量中取出一部分迭加到本次权值调整量中
- 作用：
  - 动量项反映了以前积累的调整经验，对于 t 时刻的调整起阻尼作用。
  - 当误差曲面出现骤然起伏时，可减小振荡趋势，提高训练速度。

改进2：自适应调节学习率

- 提出的原因：
  - 标准 BP 算法中，学习率 η 也称为步长，确定一个从始至终都合适的最佳学习率很难。
  - 平坦区域内，η 太小会使训练次数增加；
  - 在误差变化剧烈的区域，η 太大会因调整量过大而跨过较窄的“坑凹”处，使训练出现振荡，反而使迭代次数增加。

改进2：自适应调节学习率

- 基本思想：
  - 自适应改变学习率，使其根据环境变化增大或减小。
- 基本方法：
  - 设一初始学习率，若经过一批次权值调整后使总误差↑，则本次调整无效，且$\eta=\beta\eta(\beta<1)$；
  - 若经过一批次权值调整后使总误差↓，则本次调整有效，且$\eta=\theta\eta(\theta>1)$。

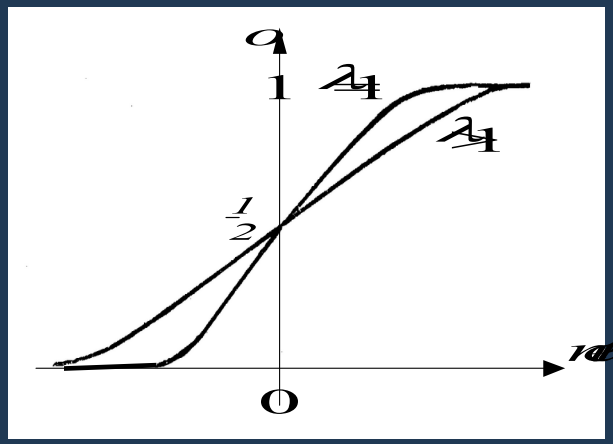
改进3：引入陡度因子

- 提出的原因：
  - 误差曲面上存在着平坦区域。
  - 权值调整进入平坦区的原因是神经元输出进入了转移函数的饱和区。
- 基本思想：
  - 如果在调整进入平坦区后，设法压缩神经元的净输入，使其输出退出转移函数的不饱和区，就可以改变误差函数的形状，从而使调整脱离平坦区。

改进 3 ： 引入陡度因子

- 基本方法：
  - 在原转移函数中引入一个陡度因子 λ

$$o = \frac{1}{1 + e^{-net/\lambda}}$$

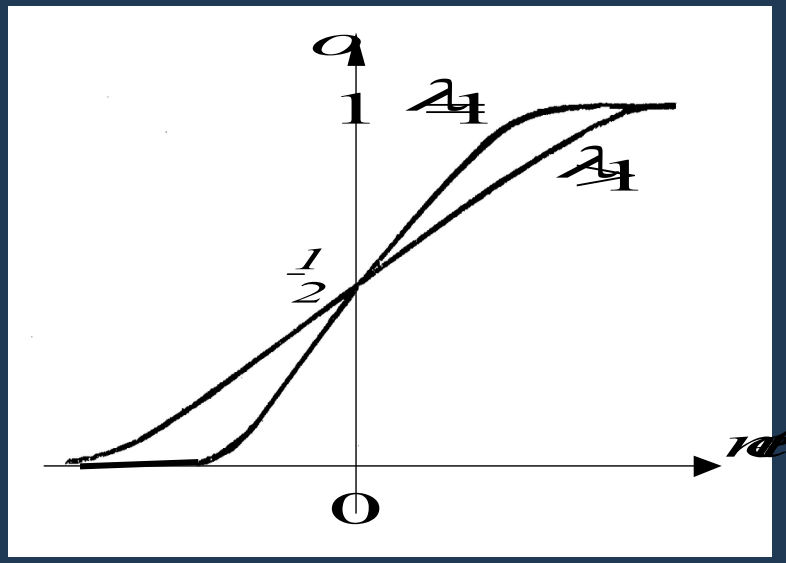


  - 当发现 $\Delta\varepsilon$ 接近零而 d-o 仍较大时，可判断已进入平坦区，此时令 λ>1 ；
  - 当退出平坦区后，再令 λ=1 。

改进 3 ： 引入陡度因子

作用分析：

- λ>1 ： *net* 坐标压缩了 λ 倍，神经元的转移函数曲线的敏感区段变长，从而可使绝对值较大的 *net* 退出饱和值。
- λ=1 ：转移函数恢复原状，对绝对值较小的 *net* 具有较高的灵敏度。
- 应用结果表明该方法对于提高 BP 算法的收敛速度十分有效。



总结

- 基于 BP 算法的多层前馈网络模型
- BP 算法的实现
  - 基本思想
  - 推导过程
  - 程序实现
- BP 学习算法的功能
- BP 学习算法的局限性
- BP 学习算法的改进

3.6 BP 网络设计基础

- 一 、训练样本集的准备
  - 1.输入输出量的选择
  - 2.输入量的提取与表示
  - 3.输出量的表示
- 二、输入输出数据的归一化
- 三、网络训练与测试

1 输出量的选择

- 输出量：代表系统要实现的功能目标
  - 系统的性能指标
  - 分类问题的类别归属
  - 非线性函数的函数值

输入量的选择

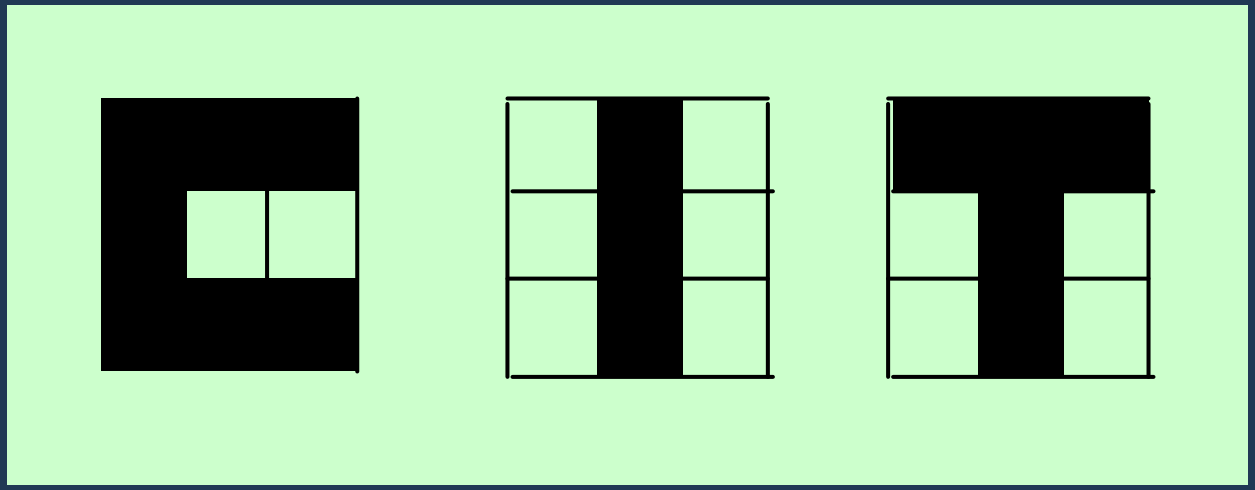
- 输入量选择的两条基本原则
  - 必须选择那些对输出影响大且能够检测或提取的变量
  - 各输入变量之间互不相关或相关性很小

输入输出量的性质

- 从输入、输出量的性质来看，可分为两类：一类是数值变量，一类是语言变量。
  - **数值变量**的值是数值确定的连续量或离散量。
  - **语言变量**是用自然语言表示的概念，其“语言值”是用自然语言表示的事物的各种属性。
    - 当选用语言变量作为网络的输入或输出变量时，需将其语言值转换为离散的数值量。

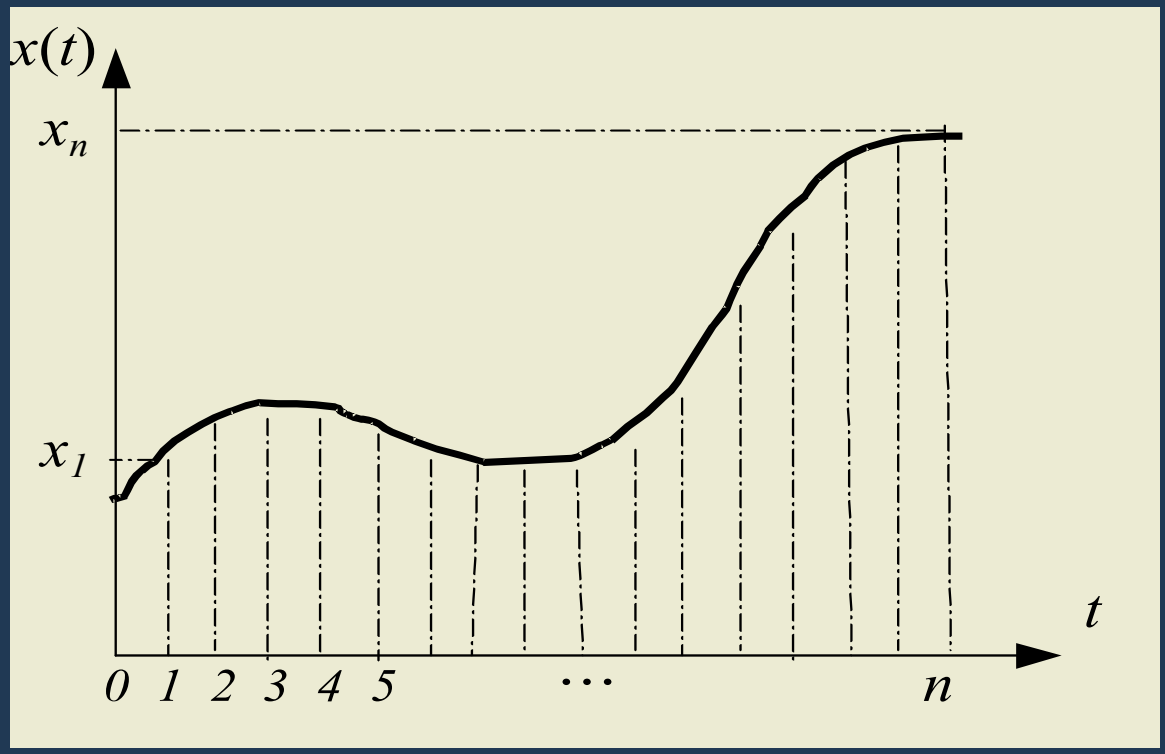
2. 输入量的提取与表示

(1) 文字符号输入



$X^C=(111100111)^T$ $X^I=(010010010)^T$ $X^T=(111010010)^T$

（2）曲线输入



$$X^p = (x_1^p, x_2^p, \ldots, x_i^p, \ldots, x_n^p)^T$$

$p=1,2,\ldots,P$

(3) 函数自变量输入

- 一般有几个输入量就设几个分量，1 个输入分量对应 1 个输入层节点。

(4) 图象输入

- 在这类应用中，一般先根据识别的具体目的从图象中提取一些有用的特征参数，再根据这些参数对输入的贡献进行筛选，这种特征提取属于图象处理的范畴。

3. 输出量的表示

(1)“*n* 中取 *1*” 表示法

“*n* 中取 *1*” 是令输出向量的分量数等于类别数，输入样本被判为哪一类，对应的输出分量取 1 ，其余 n-1 个分量全取 0 。例如，用 0001 、 0010 、 0100 和 1000 可分别表示优、良、中、差 4 个类别。

(2)“*n-1*”表示法

如果用 *n-1* 个全为 0 的输出向量表示某个类别，则可以节省一个输出节点。例如，用 000 、 001 、 010 和 100 也可表示优、良、中、差 4 个类别。

(3) 数值表示法

对于渐进式的分类，可以将语言值转化为二值之间的数值表示。数值的选择要注意保持由小到大的渐进关系，并要根据实际意义拉开距离。

二、输入输出数据的归一化

归一化也称为或标准化，是指通过变换处理将网络的输入、输出数据限制在 [0 ， 1] 或 [-1 ， 1] 区间内。

进行归一化的主要原因：

归一化的方法：

进行归一化的主要原因：

- ①网络的各个输入数据常常具有不同的***物理意义和不同的量纲***，归一化给各输入分量以同等重要的地位；
- ②BP 网的神经元均采用 Sigmoid 转移函数，变换后可防止因净输入的绝对值过大而使神经元输出***饱和***，继而使权值调整进入误差曲面的平坦区；
- ③Sigmoid 转移函数的输出在 0~1 或 -1~1 之间。***教师信号***如不进行归一化处理，势必使数值大的输出分量***绝对误差***大，数值小的输出分量绝对误差小。

归一化的方法：

将输入输出数据变换为 [0 ， 1] 区间的值常用以下变换式

$$\overline{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

其中，x_I 代表输入或输出数据，x_{min} 代表数据变化的最小值，x_{man} 代表数据的最大值。

将输入输出数据变换为 [-1 ， 1] 区间的值常用以下变换式

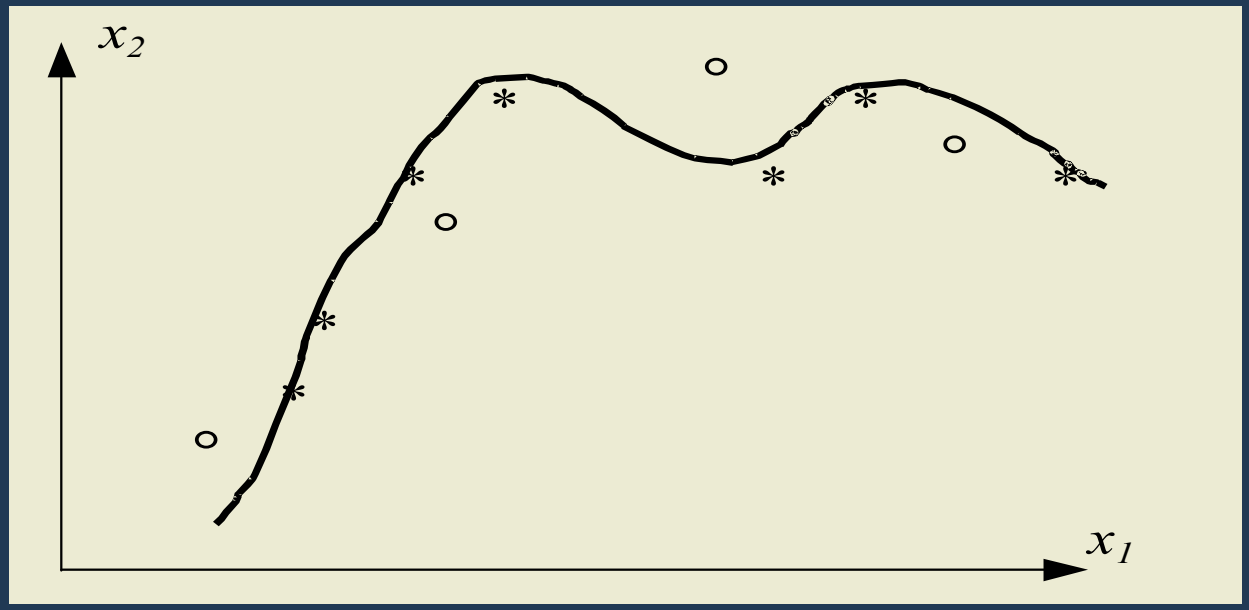
$$x_{mid} = \frac{x_{\max} + x_{\min}}{2}$$

$$\overline{x}_i = \frac{x_i - x_{mid}}{\frac{1}{2}(x_{\max} - x_{\min})}$$

其中，x_{mid} 代表数据变化范围的中间值。

三、网络训练与测试

网络的性能好坏主要看其是否具有很好的泛化能力，对泛化能力的测试不能用训练集的数据进行，而要用训练集以外的测试数据来进行检验。



在隐节点数一定的情况下，为获得好的泛化能力，存在着一个最佳训练次数。