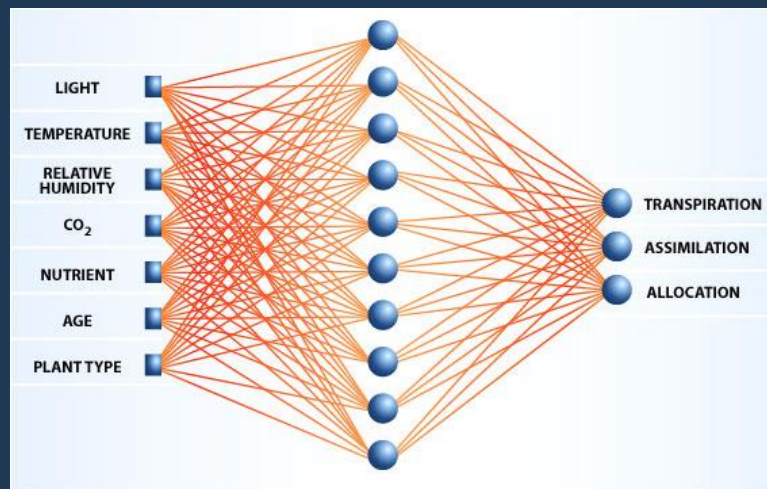


BP神经网络模型与学习算法



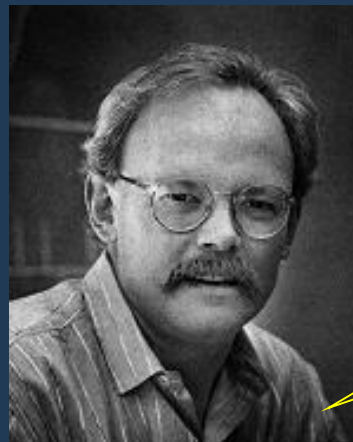
概述

- Rumelhart, McClelland于1985年提出了BP网络的误差反向传BP (Back Propagation) 学习算法

David
Rumelhart



J. McClelland

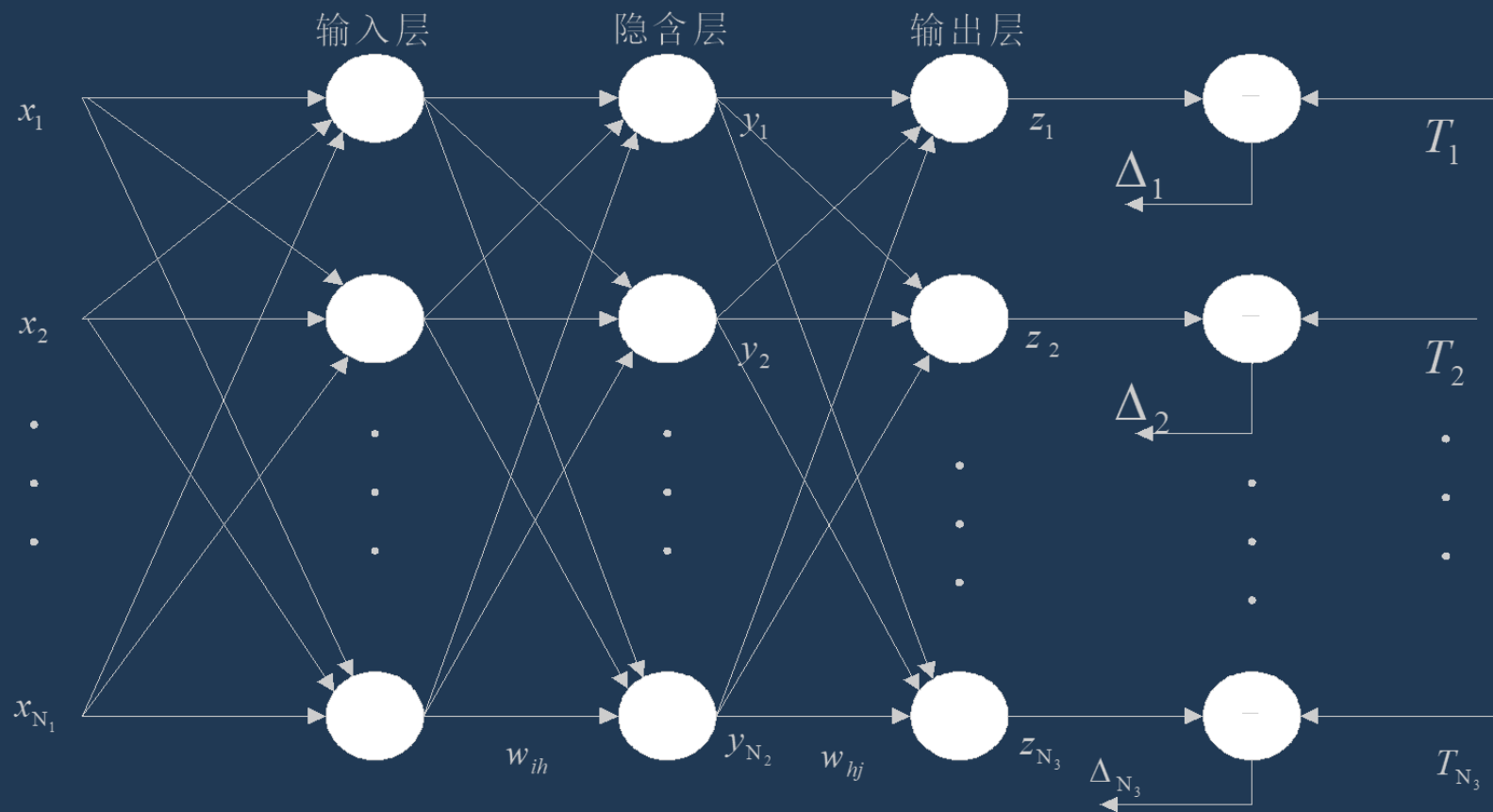


□ BP算法基本原理

- 利用输出后的误差来估计输出层的直接前导层的误差，再用这个误差估计更前一层的误差，如此一层一层的反传下去，就获得了所有其他各层的误差估计。

BP神经网络模型

□ 三层BP网络



BP神经网络模型

□ 激活函数

➤ 必须处处可导

■ 一般都使用S型函数

□ 使用S型激活函数时BP网络输入与输出关系

➤ 输入

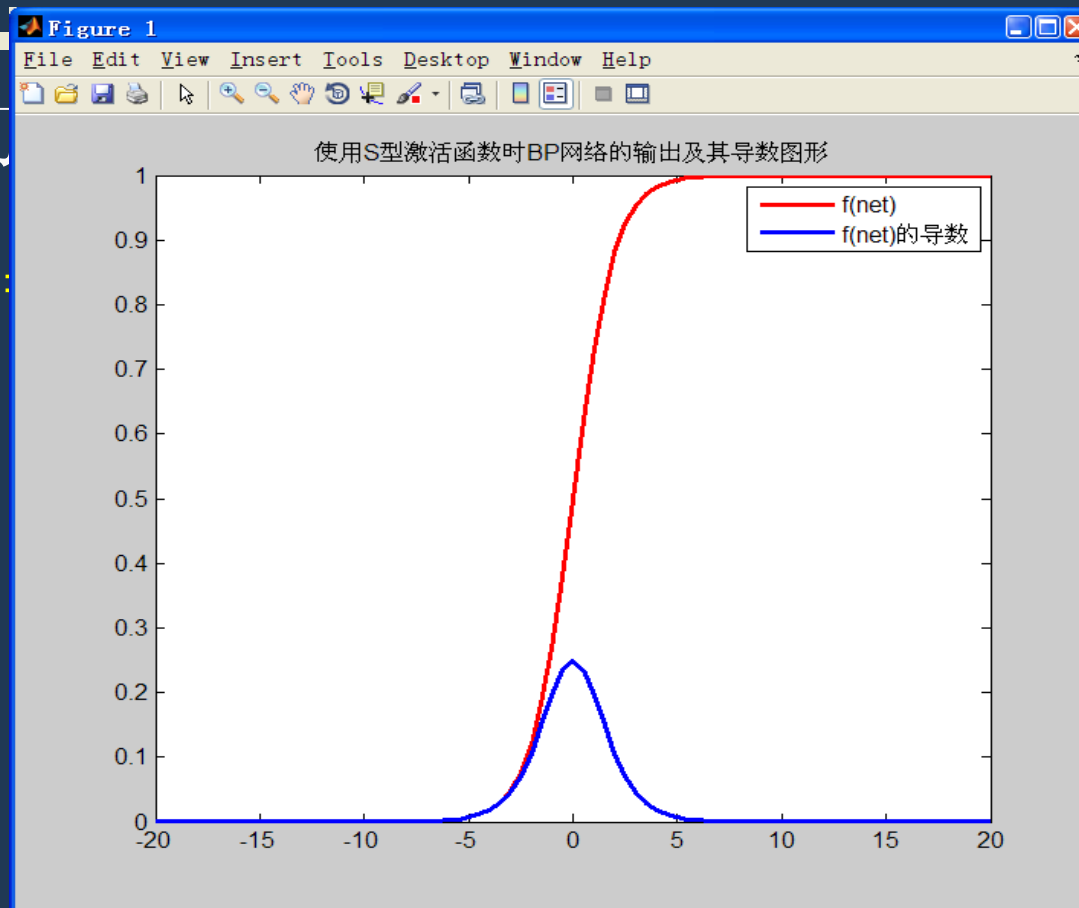
$$net = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$

➤ 输出

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

BP神经网络模型

➤ 输出的
 $f'(net)$:



➤ 根据S型激活函数的图形可知, 对神经网络进行训练, 应该将net的值尽量控制在收敛比较快的范围内

BP网络的标准学习算法

□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, 以使网络的输出不断地接近期望的输出。

□学习的本质:

- 对各连接权值的动态调整

□学习规则:


- 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网络的标准学习算法-算法思想

□学习的类型：有监督学习

□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



将误差分摊给各层的所有单元 - - - 各层单元的误差信号

修正各单元权值

□学习的过程：

➤ 信号的正向传播 → 误差的反向传播

BP网络的标准学习算法-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（监督信号）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

BP网络的标准学习算法

□网络结构

- 输入层有n个神经元，隐含层有p个神经元，输出层有q个神经元

□变量定义

- 输入向量; $x = (x_1, x_2, \dots, x_n)$
- 隐含层输入向量; $hi = (hi_1, hi_2, \dots, hi_p)$
- 隐含层输出向量; $ho = (ho_1, ho_2, \dots, ho_p)$
- 输出层输入向量; $yi = (yi_1, yi_2, \dots, yi_q)$
- 输出层输出向量; $yo = (yo_1, yo_2, \dots, yo_q)$
- 期望输出向量; $d_o = (d_1, d_2, \dots, d_q)$

BP网络的标准学习算法

- 输入层与隐含层的连接权值: w_{ih}
- 隐含层与输出层的连接权值: w_{ho}
- 隐含层各神经元的阈值: b_h
- 输出层各神经元的阈值: b_o
- 样本数据个数: $k = 1, 2, \dots, m$
- 激活函数: $f(\cdot)$
- 误差函数:
$$e = \frac{1}{2} \sum_{o=1}^q (d_o(k) - y_{o_o}(k))^2$$

BP网络的标准学习算法

□ 第一步，网络初始化

- 给各连接权值分别赋一个区间 $(-1, 1)$ 内的随机数，设定误差函数 e ，给定计算精度值 ε 和最大学习次数 M 。

□ 第二步，随机选取第 k 个输入样本及对应期望输出

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))$$

$$d_o(k) = (d_1(k), d_2(k), \dots, d_q(k))$$

BP网络的标准学习算法

□ 第三步，计算隐含层各神经元的输入和输出

$$hi_h(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h \quad h = 1, 2, \dots, p$$

$$ho_h(k) = f(hi_h(k)) \quad h = 1, 2, \dots, p$$

$$yi_o(k) = \sum_{h=1}^p w_{ho} ho_h(k) - b_o \quad o = 1, 2, \dots, q$$

$$yo_o(k) = f(yi_o(k)) \quad o = 1, 2, \dots, q$$

BP网络的标准学习算法

- 第四步，利用网络期望输出和实际输出，计算误差函数对输出层的各神经元的偏导数 $\delta_o(k)$ 。

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{ho}} \quad \frac{\partial yi_o(k)}{\partial w_{ho}} = \frac{\partial(\sum_h^p w_{ho} ho_h(k) - b_o)}{\partial w_{ho}} = ho_h(k)$$

$$\frac{\partial e}{\partial yi_o} = \frac{\partial(\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2)}{\partial yi_o} = -(d_o(k) - yo_o(k)) yo'_o(k)$$

$$= -(d_o(k) - yo_o(k)) f'(yi_o(k)) \triangleq -\delta_o(k)$$

BP网络的标准学习算法

- 第五步，利用隐含层到输出层的连接权值、输出层的 $\delta_o(k)$ 和隐含层的输出计算误差函数对隐含层各神经元的偏导数 $\delta_h(k)$ 。

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial y_{i_o}} \frac{\partial y_{i_o}}{\partial w_{ho}} = -\delta_o(k) h_{o_h}(k)$$

$$\frac{\partial e}{\partial w_{ih}} = \frac{\partial e}{\partial h_{i_h}(k)} \frac{\partial h_{i_h}(k)}{\partial w_{ih}}$$

$$\frac{\partial h_{i_h}(k)}{\partial w_{ih}} = \frac{\partial (\sum_{i=1}^n w_{ih} x_i(k) - b_h)}{\partial w_{ih}} = x_i(k)$$

BP网络的标准学习算法

$$\begin{aligned}\frac{\partial e}{\partial hi_h(k)} &= \frac{\partial(\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2)}{\partial ho_h(k)} \frac{\partial ho_h(k)}{\partial hi_h(k)} \\&= \frac{\partial(\frac{1}{2} \sum_{o=1}^q (d_o(k) - f(yi_o(k)))^2)}{\partial ho_h(k)} \frac{\partial ho_h(k)}{\partial hi_h(k)} \\&= \frac{\partial(\frac{1}{2} \sum_{o=1}^q ((d_o(k) - f(\sum_{h=1}^p w_{ho} ho_h(k) - b_o))^2))}{\partial ho_h(k)} \frac{\partial ho_h(k)}{\partial hi_h(k)} \\&= - \sum_{o=1}^q (d_o(k) - yo_o(k)) f'(yi_o(k)) w_{ho} \frac{\partial ho_h(k)}{\partial hi_h(k)} \\&= - (\sum_{o=1}^q \delta_o(k) w_{ho}) f'(hi_h(k)) \triangleq -\delta_h(k)\end{aligned}$$

BP网络的标准学习算法

- 第六步，利用输出层各神经元的 $\delta_o(k)$ 和隐含层各神经元的输出来修正连接权值 $w_{ho}(k)$ 。

$$\Delta w_{ho}(k) = -\mu \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) h o_h(k)$$

$$w_{ho}^{N+1} = w_{ho}^N + \eta \delta_o(k) h o_h(k)$$

BP网络的标准学习算法

- 第七步，利用隐含层各神经元的 $\delta_h(k)$ 和输入层各神经元的输入修正连接权。

$$\Delta w_{ih}(k) = -\mu \frac{\partial e}{\partial w_{ih}} = -\mu \frac{\partial e}{\partial hi_h(k)} \frac{\partial hi_h(k)}{\partial w_{ih}} = \delta_h(k) x_i(k)$$

$$w_{ih}^{N+1} = w_{ih}^N + \eta \delta_h(k) x_i(k)$$

BP网络的标准学习算法

□ 第八步，计算全局误差

$$E = \frac{1}{2m} \sum_{k=1}^m \sum_{o=1}^q (d_o(k) - y_o(k))^2$$

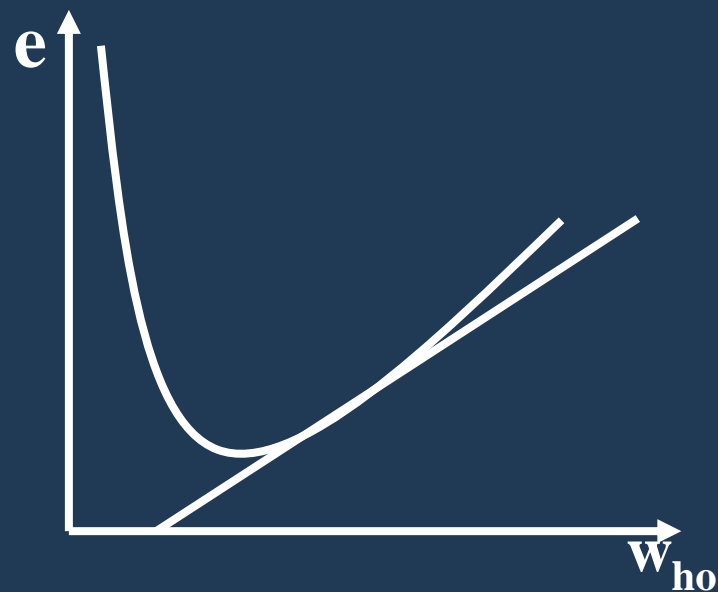
□ 第九步，判断网络误差是否满足要求。当误差达到预设精度或学习次数大于设定的最大次数，则结束算法。否则，选取下一个学习样本及对应的期望输出，返回到第三步，进入下一轮学习。

BP网络的标准学习算法

BP算法直观解释

➤ 情况一直观表达

■ 当误差对权值的偏导数大于零时，权值调整量为负，实际输出大于期望输出，权值向减少方向调整，使得实际输出与期望输出的差减少。



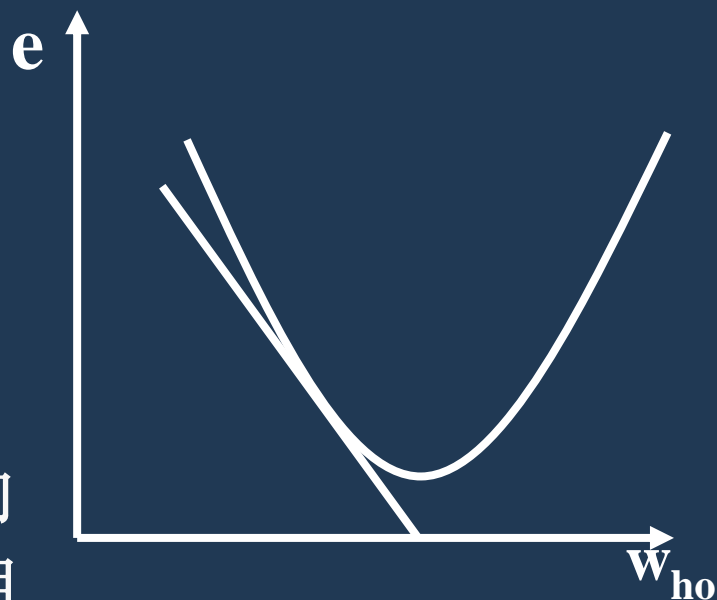
$$\frac{\partial e}{\partial w_{ho}} > 0, \text{ 此时 } \Delta w_{ho} < 0$$

BP网络的标准学习算法

□BP算法直观解释

➤情况二直观表达

■当误差对权值的偏导数小于零时，权值调整量为正，实际输出少于期望输出，权值向增大方向调整，使得实际输出与期望输出的差减少。



$$\frac{\partial e}{\partial w_{ho}} < 0, \text{ 此时 } \Delta w_{ho} > 0$$

BP神经网络的特点

□ 非线性映射能力

- 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。只要能提供足够的样本模式对供网络进行学习训练，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。

□ 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。这种能力称为泛化能力。

□ 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。