

第2章 前馈型人工神经网络

❖ M-P模型

❖ 感知机模型与学习算法

❖ 多层感知机网络

❖ 非线性连续变换单元组成的前馈网络

❖ BP算法

2.2 非线性连续变换单元组成的网络

由非线性连续变换单元组成的前馈网络，简称为BP (Back Propagation) 网络。

1. 网络的结构与数学描述

(i). 非线性连续变换单元

对于非线性连续变换单元，其输入、输出变换函数是非线性、单调上升、连续的即可。但在BP网络中，我们采用S型函数：

$$u_i = s_i = \sum_{j=1}^n w_{ij} x_j - \theta_i$$

$$y_i = f(u_i) = \frac{1}{1 + e^{-u_i}} = \frac{1}{1 + e^{-\left(\sum_{j=1}^n w_{ij} x_j - \theta_i\right)}}$$

2.2 非线性连续变换单元组成的网络

函数 $f(u)$ 是可微的，并且

$$f'(u) = \left(\frac{1}{1+e^{-u}} \right)' = f(u)(1-f(u))$$

这种函数用来区分类别

时，其结果可能是一种模

糊的概念。当 $u > 0$ 时，其

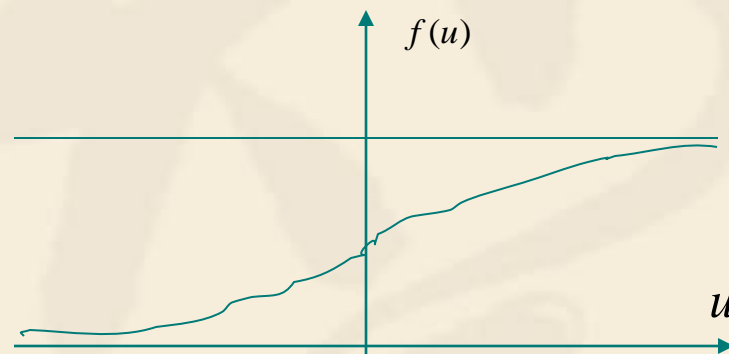
输出不是1，而是大于0.5的一个数，而当 $u < 0$

时，输出是一个小于0.5的一个数。若用这样一个

单元进行分类，当输出是0.8时，我们可认为

属于A类的隶属度（或概率）为0.8时，而

属于B类的隶属度（或概率）为0.2。



2.2 非线性连续变换单元组成的网络

(ii). 网络结构与参数

下面以四层网络为例
来介绍BP网络的结构和
参数，一般情况类似。

网络输入: $x = (x_1, x_2, \dots, x_n)^T \in R^n$

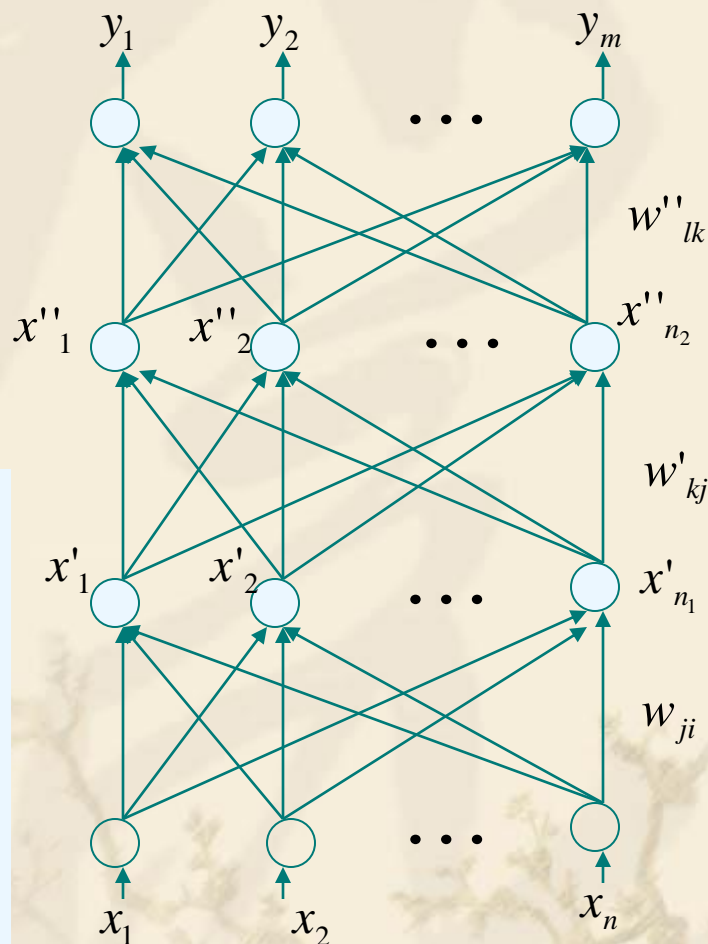
第一隐层输出: $x' = (x'_1, x'_2, \dots, x'_{n_1})^T \in R^{n_1}$

第二隐层输出: $x'' = (x''_1, x''_2, \dots, x''_{n_2})^T \in R^{n_2}$

网络输出: $y = (y_1, y_2, \dots, y_m)^T \in R^m$

连接权: $w_{ji}, w'_{kj}, w''_{lk}$

阈值: $\theta_i, \theta'_k, \theta''_l$



2.2 非线性连续变换单元组成的网络

网络的输入输出关系为：

$$\begin{cases} x'_j = f\left(\sum_{i=1}^n w_{ji} x_i - \theta_j\right), & j = 1, 2, \dots, n_1 \\ x''_k = f\left(\sum_{j=1}^{n_1} w'_{kj} x'_j - \theta'_k\right), & k = 1, 2, \dots, n_2 \\ y_l = f\left(\sum_{j=1}^{n_2} w''_{lk} x''_k - \theta''_l\right), & l = 1, 2, \dots, m \end{cases}$$

显然可以将阈值归入为特别的权，从而网络的参数可用 W 表示（ W 为一个集合）。上述网络实现了一个多元连续映射：

$$y = F(x, W): R^n \rightarrow R^m$$

2.2 非线性连续变换单元组成的网络

(iii). 网络的学习问题

学习的目标：通过网络（或 $F(x, W)$ ）来逼近一个连续系统，即连续变换函数 $G(x)$ 。

学习的条件：一组样本（对）

$$S = \{(x^1, y^1), (x^2, y^2), \dots, (x^N, y^N)\}$$

对于样本对 (x^i, y^i) ，存在 W^i 使得

$$y^i = F(x^i, W), \quad W \in \mathbf{W}^i \subset R^p, \quad p = n \times n_1 + n_1 + n_1 \times n_2 + n_2 + n_2 \times m + m$$

对于所有样本的解空间为：

$$\mathbf{W} = \bigcap_{i=1}^N \mathbf{W}^i$$

2.2 非线性连续变换单元组成的网络

(iv). Kolmogorov定理

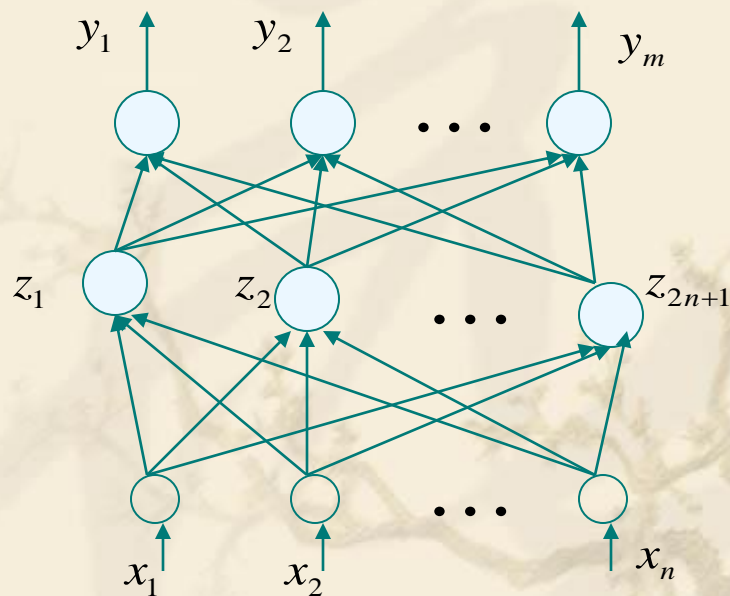
Kolmogorov定理 (映射神经网络存在定理, 1950s)

给定任何连续函数 $f: [0,1]^n \rightarrow R^m$, $y = f(x)$, 则 f 能够被一个三层前馈神经网络所实现, 其中网络的隐单元数为 $2n+1$ 。

$$z_j = \sum_{i=1}^n \lambda^j \psi(x_i + j\varepsilon) + j, \quad j=1,2,\dots,2n+1$$

$$y_k = \sum_{j=1}^{2n+1} g_k(z_j), \quad k=1,2,\dots,m$$

其中 ψ 为连续单调递增函数, g_j 为连续函数, λ 为常数, ε 为正有理数。



注意：定理未解决构造问题。

2.2 非线性连续变换单元组成的网络

2. BP学习算法

(i). 基本思想

BP算法属于 δ 学习律，是一种有监督学习：

样本输入： $x^1, x^2, \dots, x^N \Rightarrow t^1, t^2, \dots, t^N$ (理想输出或导师值)

\Downarrow

网络实际输出： $y^1, y^2, \dots, y^N \Rightarrow Error$ (误差)

对于辅助变量并将阈值归入权参数：

$$x_0 \equiv -1, w_{j0} = \theta_j, x'_0 \equiv -1, w'_{k0} = \theta'_k, x''_0 \equiv -1, w''_{l0} = \theta''_l$$

则有：

$$x'_j = f\left(\sum_{i=0}^n w_{ji} x_i\right), \quad x''_k = f\left(\sum_{j=0}^{n_1} w'_{kj} x'_j\right), \quad y_l = f\left(\sum_{k=0}^{n_2} w''_{lk} x''_k\right)$$

2.2 非线性连续变换单元组成的网络

考虑第 μ 个样本的误差：

$$E_{\mu} = \frac{1}{2} \| t^{\mu} - y^{\mu} \|^2 = \frac{1}{2} \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu})^2$$

进一步得总误差：

$$E = \sum_{\mu=1}^N E_{\mu} = \frac{1}{2} \sum_{\mu=1}^N \| t^{\mu} - y^{\mu} \|^2 = \frac{1}{2} \sum_{\mu=1}^N \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu})^2$$

引入权参数矩阵：

$$\mathbf{W} = (w_{ji})_{n_1 \times (n+1)}, \quad \mathbf{W}' = (w'_{kj})_{n_2 \times (n_1+1)}, \quad \mathbf{W}'' = (w''_{lk})_{m \times (n_2+1)}$$

和总权参数向量：

$$W = \begin{bmatrix} \text{vec}[\mathbf{W}] \\ \text{vec}[\mathbf{W}'] \\ \text{vec}[\mathbf{W}''] \end{bmatrix} = (w_{10}, w_{11}, \dots, w_{sg}, \dots, w_{cd})^T$$

2.2 非线性连续变换单元组成的网络

根据总误差得到一般性的梯度算法：

$$E = \sum_{\mu=1}^N E_{\mu}(W, t^{\mu}, x^{\mu})$$
$$\Delta w_{sg} = -\eta \frac{\partial E}{\partial w_{sg}} = -\eta \sum_{\mu=1}^N \frac{\partial E_{\mu}(W, t^{\mu}, x^{\mu})}{\partial w_{sg}}$$
$$\Delta E = \sum_{s,g} \frac{\partial E}{\partial w_{sg}} \Delta w_{sg} = -\eta \sum_{s,g} \left(\frac{\partial E}{\partial w_{sg}} \right)^2$$

终止规则： $\Delta E = 0$, $E \approx 0$? $E \leq \varepsilon (> 0)$

这里用梯度法可以使总的误差向减小的方向变化，直到 ΔE 或梯度为零结束。这种学习方式使权向量 W 达到一个稳定解，但无法保证 E 达到全局最优，一般收敛到一个局部极小解。

2.2 非线性连续变换单元组成的网络

(ii). BP算法的推导

令 n_0 为迭代次数，则得一般性梯度下降法：

$$\begin{cases} w''_{lk}(n_0 + 1) = w''_{lk}(n_0) - \eta \frac{\partial E}{\partial w''_{lk}} \\ w'_{kj}(n_0 + 1) = w'_{kj}(n_0) - \eta \frac{\partial E}{\partial w'_{kj}} \\ w_{ji}(n_0 + 1) = w_{ji}(n_0) - \eta \frac{\partial E}{\partial w_{ji}} \end{cases}$$

其中 η 为学习率，是一个大于零的较小的实数。

先考虑对于 w''_{lk} 的偏导数：

$$\frac{\partial E}{\partial w''_{lk}} = \sum_{\mu=1}^N \frac{\partial E_{\mu}}{\partial y_l^{\mu}} \frac{\partial y_l^{\mu}}{\partial u''_l^{\mu}} \frac{\partial u''_l^{\mu}}{\partial w''_{lk}} = - \sum_{\mu=1}^N (t_l^{\mu} - y_l^{\mu}) f'(u''_l^{\mu}) x''_k^{\mu}$$
$$(u''_l^{\mu} = \sum_{k=0}^{n_2} w''_{lk} x''_k^{\mu})$$

2.2 非线性连续变换单元组成的网络

在上式中, x''_k^μ 为第 μ 个样本输入网络时, x'_k 的对应值。另外

$$f'(u''_l^\mu) = f(u''_l^\mu)(1 - f(u''_l^\mu)) = y_l^\mu(1 - y_l^\mu)$$

令 $\delta_l^\mu = (t_l^\mu - y_l^\mu)y_l^\mu(1 - y_l^\mu)$

则: $w''_{lk}(n_0 + 1) = w''_{lk}(n_0) - \eta \frac{\partial E}{\partial w''_{lk}} = w''_{lk}(n_0) + \eta \sum_{\mu=1}^N \delta_l^\mu x''_k^\mu$

为了方便, 引入记号:

$$\begin{cases} y_l = f(u'_l), & u'_l = \sum_{k=0}^{n_2} w'_{lk} x'_k \\ x''_k = f(u'_k), & u'_k = \sum_{j=0}^{n_1} w'_{kj} x'_j \\ x'_j = f(u_j), & u_j = \sum_{i=0}^n w_{ji} x_i \end{cases}$$

2.2 非线性连续变换单元组成的网络

对于 w'_{kj} 的偏导数，我们有：

$$\begin{aligned}\frac{\partial E}{\partial w'_{kj}} &= \sum_{\mu=1}^N \sum_{l=1}^m \frac{\partial E_{\mu}}{\partial y_l^{\mu}} \frac{\partial y_l^{\mu}}{\partial u''_l^{\mu}} \frac{\partial u''_l^{\mu}}{\partial x''_k^{\mu}} \frac{\partial x''_k^{\mu}}{\partial u''_k^{\mu}} \frac{\partial u''_k^{\mu}}{\partial w'_{kj}} \\&= - \sum_{\mu=1}^N \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu}) f'(u''_l^{\mu}) w''_{lk} f'(u''_k^{\mu}) x''_j^{\mu} \\&= - \sum_{\mu=1}^N \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu}) y_l^{\mu} (1 - y_l^{\mu}) w''_{lk} x''_k^{\mu} (1 - x''_k^{\mu}) x''_j^{\mu} \\&= \sum_{\mu=1}^N \sum_{l=1}^m \delta_l^{\mu} w''_{lk} x''_k^{\mu} (1 - x''_k^{\mu}) x''_j^{\mu} \\&= \sum_{\mu=1}^N \dot{\delta}_k^{\mu} x''_j^{\mu}\end{aligned}$$

其中 $\dot{\delta}_k^{\mu} = \sum_{l=1}^m \delta_l^{\mu} w''_{lk} x''_k^{\mu} (1 - x''_k^{\mu}) = x''_k^{\mu} (1 - x''_k^{\mu}) \sum_{l=1}^m \delta_l^{\mu} w''_{lk}$

2.2 非线性连续变换单元组成的网络

这样我们有：

$$w'_{kj}(n_0 + 1) = w'_{kj}(n_0) + \eta \sum_{\mu=1}^N \dot{\delta}_k^{\mu} x'^{\mu}_j$$

类似的推导可得：

$$w_{ji}(n_0 + 1) = w_{ji}(n_0) + \eta \sum_{\mu=1}^N \ddot{\delta}_j^{\mu} x_i^{\mu}$$

$$\text{其中 } \ddot{\delta}_j^{\mu} = \sum_{k=1}^{n_1} w'_{kj} \dot{\delta}_k^{\mu} \quad x'^{\mu}_j (1 - x'^{\mu}_j) = x'^{\mu}_j (1 - x'^{\mu}_j) \sum_{k=1}^{n_1} w'_{kj} \dot{\delta}_k^{\mu}$$

(iii). BP算法

Step 1. 赋予初值： $w_{sg}(0) = \alpha(\text{Random}(\bullet) - 0.5)$, $\alpha > 0$, $(w_{sg} \rightarrow w_{ji}, w'_{kj}, w''_{lk})$

Step 2. 在 n_0 时刻，计算 x'^{μ}_j , x''^{μ}_k , y_l^{μ} 及其广义误差

$$\delta_l^{\mu}, l = 1, 2, \dots, m; \quad \dot{\delta}_k^{\mu}, k = 1, 2, \dots, n_2; \quad \ddot{\delta}_j^{\mu}, j = 1, 2, \dots, n_1$$

2.2 非线性连续变换单元组成的网络

Step 3. 修正权值：

$$w''_{lk}(n_0 + 1) = w''_{lk}(n_0) + \eta \sum_{\mu=1}^N \delta_l^\mu x''_{k\mu}$$

$$w'_{kj}(n_0 + 1) = w'_{kj}(n_0) + \eta \sum_{\mu=1}^N \dot{\delta}_k^\mu x'_{j\mu}$$

$$w_{ji}(n_0 + 1) = w_{ji}(n_0) + \eta \sum_{\mu=1}^N \ddot{\delta}_j^\mu x_i^\mu$$

Step 4. 计算修正后的误差：

$$E(n_0 + 1) = \sum_{\mu=1}^N E_\mu(\mathbf{W}(n_0 + 1), t^\mu, x^\mu)$$

若 $E(n_0 + 1) < \varepsilon$ ($\varepsilon > 0$, 预先给定) 或 $\left| \frac{\partial E}{\partial w_{sg}} \right| < \varepsilon$ ，算法结束，
否则返回到Step 2。

2.2 非线性连续变换单元组成的网络

BP算法的讨论：a). 这里的梯度是对于全部样本求的，因此是一种批处理算法，即 Batch-way, 它符合梯度算法，稳定地收敛到总误差的一个极小点而结束。（注意：按总误差小于 ε 可能导致算法不收敛。）b). 实际中更常用的是对每个样本修改，即自适应算法，当每次样本是随机选取时，可通过随机逼近理论证明该算法也是收敛的。特点是收敛速度快。C). 为了使得算法既稳定，又具有快的收敛速度，可以使用批处理与自适应相补充的算法，即选取一组样本（远小于全部样本）进行计算梯度并进行修正，其它不变。

2.2 非线性连续变换单元组成的网络

3. BP网络误差曲面的特性

BP网络的误差公式为：

$$E = \frac{1}{2} \sum_{\mu=1}^N \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu})^2$$

$y_l = f(u_l)$ 是一种非线性函数，而多层的BP网络中 u_l 又是上一层神经元状态的非线性函数，用 E_{μ} 表示其中一个样本对应的误差，则有：

$$E_{\mu} = \frac{1}{2} \sum_{l=1}^m (t_l^{\mu} - y_l^{\mu})^2 = E(\mathbf{W}, t^{\mu}, x^{\mu})$$

$$E = \sum_{\mu=1}^N E(\mathbf{W}, t^{\mu}, x^{\mu})$$

可见， E 与 \mathbf{W} 有关，同时也与所有样本对有关，即与 $S = \{(x^1, y^1), (x^2, y^2), \dots, (x^N, y^N)\}$ 有关。

2.2 非线性连续变换单元组成的网络

假定样本集 S 给定，那么 E 是 \mathbf{W} 的函数。在前面考虑的4层网络中，权值参数的总个数为：

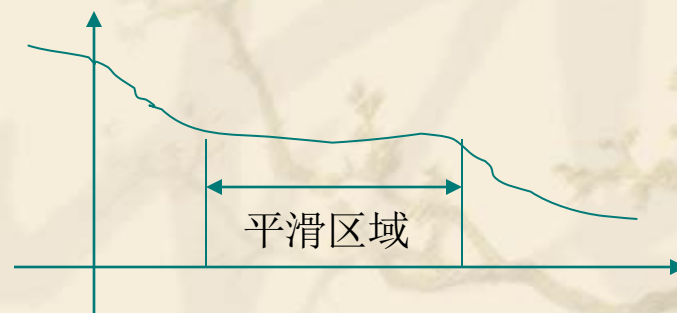
$$n_{\mathbf{W}} = n(n_1 + 1) + n_1(n_2 + 1) + n_2(m + 1)$$

那么在加上 E 这一维数，在 $n_{\mathbf{W}} + 1$ 维空间中， E 是一个具有极其复杂形状的曲面。如果在考虑样本，其形状就更为复杂，难于想象。从实践和理论上，人们得出了下面三个性质：

(i). 平滑区域

$$\delta_l^\mu = (t_l^\mu - y_l^\mu) y_l^\mu (1 - y_l^\mu)$$

广义误差 \neq 误差



2.2 非线性连续变换单元组成的网络

(ii). 全局最优解 \mathbf{W}^* 不唯一

\mathbf{W}^* 中的某些元素进行置换依然是全局最优解，这从右边的简单模型可以看出。

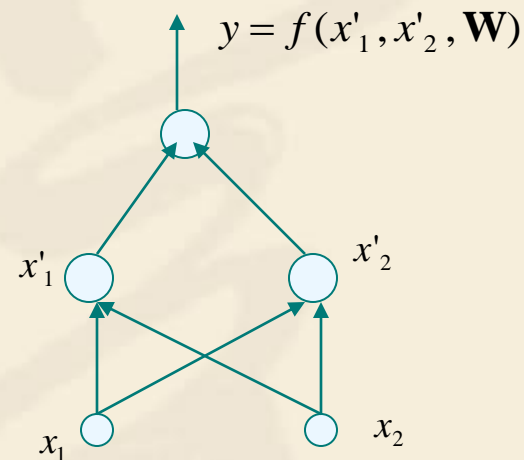
(iii). 局部极小

一般情况下，BP算法会收敛到一个局部极小解，即：

$$\mathbf{W}(n_0) \rightarrow \mathbf{W}^0$$

当 $E(\mathbf{W}^0) < \varepsilon$ ，算法以希望误差收敛；

当 $E(\mathbf{W}^0) \geq \varepsilon$ ，算法不以希望误差收敛，但可按梯度绝对值小于预定值结束。



2.2 非线性连续变换单元组成的网络

4. 算法的改进

(i). 变步长算法 (η 是由一维搜索求得)

Step 1. 赋予初始权值 $\mathbf{W}(0)$ 和允许误差 $\varepsilon > 0$;

Step 2. 在时刻 n_0 , 计算误差 $E(\mathbf{W}(n_0))$ 的负梯度
(方向) : $d^{(n_0)} = -\nabla E(\mathbf{W})|_{\mathbf{W}=\mathbf{W}(n_0)}$

Step 3. 若 $\|d^{(n_0)}\| < \varepsilon$, 结束; 否则从 $\mathbf{W}(n_0)$ 出发,
沿 $d^{(n_0)}$ 做一维搜索, 求出最优步长 $\eta(n_0)$:

$$\eta(n_0) = \arg \min_{\eta} E(\mathbf{W}(n_0) + \eta d^{(n_0)})$$

Step 4. $\mathbf{W}(n_0 + 1) = \mathbf{W}(n_0) + \eta(n_0)d^{(n_0)}$, 转 Step 2。

2.2 非线性连续变换单元组成的网络

步长（学习率） $\eta(n_0)$ 的确定方法：

(a). 求最优解：对 η 求导数，并令其为零，直接求解：

$$\frac{\partial E(W(n_0) + \eta d^{(n_0)})}{\partial \eta} = 0$$

(b). 迭代修正法：令 $\Delta E = E(W(n_0) + \eta d^{(n_0)}) - E(W(n_0))$

$$\eta^{new} = \begin{cases} \eta^{old} \varphi, & \text{if } \Delta E < 0 \\ \eta^{old} \beta, & \text{if } \Delta E > 0 \end{cases}$$

其中 $\varphi = 1 + \delta$, $\beta = 1 - \delta$, $\delta > 0$

2.2 非线性连续变换单元组成的网络

(ii). 加动量项

为了防止震荡并加速收敛，可采用下述规则：

$$\begin{aligned}\mathbf{W}(n_0 + 1) &= \mathbf{W}(n_0) + \eta(n_0)d^{(n_0)} + \underline{\alpha\Delta\mathbf{W}(n_0)} \\ &= \mathbf{W}(n_0) + \eta(n_0)\left(d^{(n_0)} + \frac{\alpha\eta(n_0 - 1)}{\eta(n_0)}d^{(n_0 - 1)}\right)\end{aligned}$$

其中 $\alpha\Delta\mathbf{W}(n_0)$ 为动量项 ($\Delta\mathbf{W}(n_0) = \mathbf{W}(n_0) - \mathbf{W}(n_0 - 1)$, $0 < \alpha < 1$)

注意：上式类似于共轭梯度法的算式，但是这里

$d^{(n_0)}, d^{(n_0 - 1)}$ 不共轭。因此可能出现误差增加的现象，即 $\Delta E > 0$ ，这时可令 $\alpha = 0$ ，即退回到原来的梯度算法。

2.2 非线性连续变换单元组成的网络

(iii). 加入 γ 因子

当算法进入平坦区，即 $(1 - y_l^\mu) y_l^\mu \approx 0$ ，则 $|u_l^{\prime\prime\mu}| \rightarrow +\infty$ 。

为了消除或减弱这种现象，引入 γ 因子，使得：

$$y_l^\mu = \frac{1}{1 + \exp(-u_l^{\prime\prime\mu})}, \quad u_l^{\prime\prime\mu} = \sum_{k=0}^{n_2} w_{lk}^{\prime\prime} x_k^{\prime\prime\mu} / \gamma_l^\mu, \quad \gamma_l^\mu > 1$$

(iv). 模拟退火方法

在所有权上加一个噪声，改变误差曲面的形状，使用模拟退火的机制，使算法逃离局部极小点，达到全局最优而结束。

2.2 非线性连续变换单元组成的网络

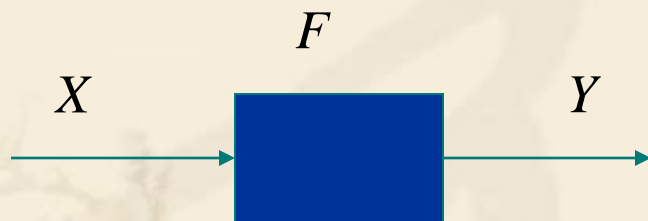
5. BP网络的设计

(i). 输入输出层的设计

BP网络输入、输出层单元个数是完全根据实际问题来设计的，我们分三种情况讨论：

A. 系统识别

$$y = F(X): R^n \rightarrow R^m$$



这时输入单元个数为 n ；

输出单元个数为 m 。

2.2 非线性连续变换单元组成的网络

B. 分类问题

$$S = \{(x^1, t^1), (x^2, t^2), \dots, (x^N, t^N)\}, \quad t^i \in \{C^1, C^2, \dots, C^m\}$$

(a). 若 $t^i \leftrightarrow C^j$, 则令 $t^i = \lambda j$ ($\lambda > 0$), 这样输出层仅需要一个单元。

(b). 若 $t^i \leftrightarrow C^j$, 则令:

$$t^i = (0, \dots, 0, 1, 0, \dots, 0)^T \quad (\text{第 } j \text{ 个分量为 } 1, \text{ 其余分量为 } 0)$$

这样输出层则需要 m 个单元。

(c). 二进制编码方法

对 C^1, C^2, \dots, C^m 进行二进制编码, 编码位数为

2.2 非线性连续变换单元组成的网络

$\log_2 m$ ，这样输出层仅需 $\log_2 m$ 个单元。

(ii). 隐单元数与映射定理

1989年，R. Hecht-Nielson证明了任何一个闭区间内的连续函数都可以用一个三层（仅有一个隐层）BP网络来逼近（任意给定精度）。

引理2.1 任意给定一个连续函数 $g \in C(a,b)$ 及精度 $\varepsilon > 0$ ，必存在一个多项式 $p(x)$ ，使得不等式 $|g(x) - p(x)| < \varepsilon$ 对任意 $x \in [a,b]$ 成立。

引理2.2 任意给定一个周期为 2π 的连续函数

$g \in C_{2\pi}$ 及精度 $\varepsilon > 0$ ，必存在一个三角函数多项式 $T(x)$ ，使得 $|g(x) - T(x)| < \varepsilon$ 对于 $\forall x \in R$ 成立。

2.2 非线性连续变换单元组成的网络

在 n 维空间中，任一向量 x 都可表示为

$$x = c_1 e_1 + c_2 e_2 + \cdots + c_n e_n$$

其中 $\{e_1, e_2, \cdots, e_n\}$ 为 R^n 的一个正交基。同样考虑连续函数空间 $C[a, b]$ 或 $C_{2\pi}$ ，必然存在一组正交函数序列 $\{\varphi_k(x)\}_{k=1}^{\infty}$ ，那么对 $\forall g(x) \in C[a, b]$ ，则

$$g(x) = \sum_{k=1}^{\infty} c_k \varphi_k(x) = \sum_{k=1}^N c_k \varphi_k(x) + \varepsilon_N(x)$$

或对 $\forall g_F(x) \in C_{2\pi}$ ，则有

$$g_F(x) = \sum_k c_k e^{2\pi i k x} = \sum_{k=-\infty}^{+\infty} c_k e^{2\pi i k x} = \sum_{k=-N}^N c_k e^{2\pi i k x} + \varepsilon_N(x)$$

其中 $c_k = \int g(x) e^{-2\pi i k x} dx$ 为傅里叶系数。

2.2 非线性连续变换单元组成的网络

当 N 充分大时, 对每个 x 成立:

$$g_F^N(x) = \sum_{k=-N}^N c_k e^{2\pi i k x}$$

$$|g_F(x) - g_F^N(x)| < \varepsilon (> 0)$$

进一步考虑 $c([0,1]^n)$ 中的多元连续函数:

$$g(x): [0,1]^n \rightarrow R, \quad g(x) \in c([0,1]^n)$$

根据傅立叶级数展开理论, 若

$$\int_{[0,1]^n} |g(x)| dx_1 \cdots dx_n < \infty$$

则同样存在一个 N 步傅立叶级数和函数:

$$g_F(x, N, g) = \sum_{k_1=-N}^N \cdots \sum_{k_n=-N}^N c_{k_1 \cdots k_n} e^{2\pi i \sum_{j=1}^n k_j x_j} = \sum_{K=(-N, \cdots, -N)}^{(N, \cdots, N)} c_{k_1 \cdots k_n} e^{2\pi i K^T x}$$

2.2 非线性连续变换单元组成的网络

其中系数为：
$$c_{k_1 \dots k_n} = \int_{[0,1]^n} g(x) e^{-2\pi i K^T x} dx$$

并且当 $N \rightarrow \infty$ 时，满足

$$g_F(x, N, g) \rightarrow g(x)$$

即 $g_F(x, \infty, g)$ 在 $[0,1]^n$ 可以完全收敛达到 $g(x)$ 。

现在考虑对一个任意连续映射： $h(x):[0,1]^n \rightarrow R^m$

其中 $h(x)=[h_1(x), \dots, h_n(x)]$, $h_j(x) \in c([0,1]^n)$ ，则 $h(x)$ 的

每个分量也都可以用上面的傅立叶级数表示，
依此就可以得到下面的影射定理（定理中所考虑的三层网络输出单元为线性单元）。

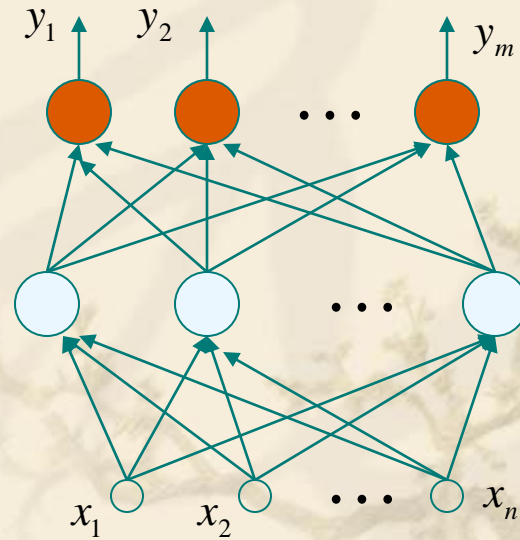
2.2 非线性连续变换单元组成的网络

映射定理(Hecht-Nielsen): 给定任意精度 $\varepsilon > 0$,
对于一个连续映射 $h(x): [0,1]^n \rightarrow R^m$, 其中:

$$\int_{[0,1]^n} \|h(x)\| dx_1 \cdots dx_n < \infty$$

那么必存在在一个三层BP神经网络来逼近函数,
使得在每点上的误差不超过 ε 。

证明: 由于输出单元是独立的,
分别与 $h(x)$ 的每个分量
函数相对应, 我们仅需要
对单个输出单元和分量函
数来证明。



2.2 非线性连续变换单元组成的网络

根据傅立叶级数理论，对于 $h(x)$ 的分量 $h_j(x)$ ，则

$$|h_j(x) - g_F(x, N, h_j)| < \delta_1 (> 0), \quad \forall x \in [0, 1]^n$$

其中 $g_F(x, N, h_j)$ 是 $h_j(x)$ 的 N 步傅立叶级数和函数：

$$g_F(x, N, h_j) = \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} c_{k_1, k_2, \dots, k_n} e^{2\pi i K^T x}, \quad c_{k_1 \dots k_n} = \int_{[0, 1]^n} h_j(x) e^{-2\pi i K^T x} dx$$

下面证明傅立叶级数中任意三角函数可以用三层BP子网络来逼近，那么通过傅立叶级数的线性组合就可以保证用三层BP网络来逼近函数 $h_j(x)$ 。

考虑结构为 $n - n_1 - 1$ 的三层BP网络，其输出为：

2.2 非线性连续变换单元组成的网络

$$y = \sum_{j=1}^{n_1} w_j f\left(\sum_{k=1}^n w_{jk} x_k - \theta_j\right)$$

我们来证明输出函数 y 能够逼近任何三角函数：

令

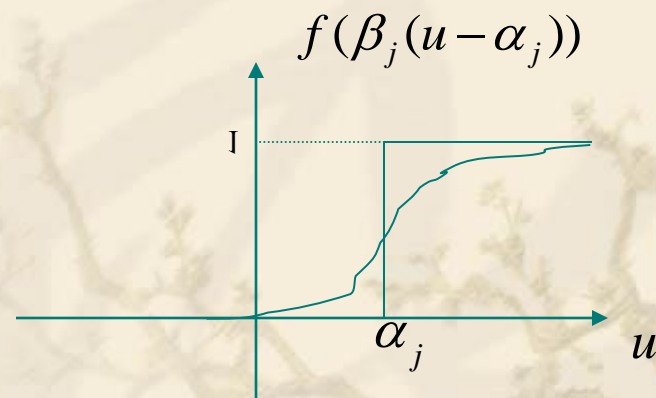
$$\sin(2\pi K^T x) = \sin(u) \quad (u = 2\pi K^T x = 2\pi \sum_{l=1}^n k_l x_l \in [d, e])$$

$$\sum_{k=1}^{n_1} w_{jk} x_k - \theta_j = u'_j - \theta_j = \beta_j(u - \alpha_j)$$

考虑函数 $f(\beta_j(u - \alpha_j))$ ，当 $\beta_j \rightarrow +\infty$ ，趋向于单位阶跃函数（见右图），则

$$S(\alpha, \beta, W, u) = \sum_{j=1}^{n_1} w_j f(\beta_j(u - \alpha_j))$$

为一些近似单位阶跃函数的线性叠加，故当 n_1 充分



2.2 非线性连续变换单元组成的网络

大时，我们可将区间 $[d, e]$ 充分的细分，选取 α_j 和 β_j ，使得 $|S(\alpha, \beta, W, u) - \sin(u)| < \delta_2$ (> 0)，或

$$\left| \sum_{j=1}^{n_1} w_j f\left(\sum_{k=1}^n w_{jk} x_k - \theta_j\right) - \sin(2\pi K^T x) \right| < \delta_2$$

即得：

$$\sum_{j=1}^{n_1} w_j f\left(\sum_{k=1}^n w_{jk} x_k - \theta_j\right) \approx \sin(2\pi K^T x)$$

对于 $h_j(x)$ ，我们有下面的展开：

2.2 非线性连续变换单元组成的网络

$$\begin{aligned}h_j(x) &\approx g_F(x, N, h_j) = \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} c_K e^{2\pi i K^T x} \\&= \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} c_K (\sin(2\pi K^T x) + i \cos(2\pi K^T x)) \\&= \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} a_K \sin(2\pi K^T x) + b_K \cos(2\pi K^T x) \\&\quad + i \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} a'_K \sin(2\pi K^T x) + b'_K \cos(2\pi K^T x) \\&\quad \searrow \\&\quad 0\end{aligned}$$

2.2 非线性连续变换单元组成的网络

使用充分多的隐单元，可得

$$y(x) = \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} a_K S(\alpha_K, \beta_K, W_K, u_K) + b_K S(\alpha'_K, \beta'_K, W'_K, u_K)$$

令

$$h_F(x) = \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} a_K \sin(2\pi K^T x) + b_K \cos(2\pi K^T x)$$

$$\begin{aligned} |h_j(x) - y(x)| &= |h_j(x) - h_F(x) + h_F(x) - y(x)| \\ &\leq |h_j(x) - h_F(x)| + |h_F(x) - y(x)| \\ &\leq |h_j(x) - h_F(x)| + \left| \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} a_K (\sin(u_K) - S(\alpha_K, \beta_K, W_K, u_K)) + b_K (\cos(u_K) - S(\alpha'_K, \beta'_K, W'_K, u_K)) \right| \\ &\leq \delta_1 + \delta_2 \sum_{K=(-N, \dots, -N)}^{(N, \dots, N)} |a_K| + |b_K| \leq \varepsilon \quad (\forall x \in [0, 1]^n) \end{aligned}$$

证毕

2.2 非线性连续变换单元组成的网络

(iii). 隐单元数的选择

隐单元数：小，结构简单，逼近能力差，不收敛；大，结构复杂，逼近能力强，收敛慢。

对于用作分类的三层BP网络，可参照多层感知机网络的情况，得到下面设计方法：

$$(a). \quad N < \sum_{i=0}^n \binom{n_1}{i}, \quad (i > n_1, \binom{n_1}{i} = 0)$$

其中 N 为样本个数，选取满足上式最小的 n_1 。

$$(b). \quad n_1 = \sqrt{n+m} + a \quad (a \in \{1, 2, \dots, 10\})$$

$$(c). \quad n_1 = \log_2 n$$

2.2 非线性连续变换单元组成的网络

(iv). 网络参数初始值的选取

初试权：随机， 比较小（接近于0）， 保证状态值较小， 不在平滑区域内。

6. BP网络的应用

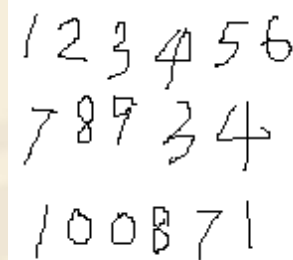
(i). 模式识别、分类。用于语音、文字、图象的识别，用于医学图象的分类、诊断等。

(ii). 函数逼近与系统建模。用于非线性系统的建模，拟合非线性控制曲线，机器人的轨迹控制，金融预测等。

2.2 非线性连续变换单元组成的网络

(iii). 数据压缩。在通信中的编码压缩和恢复，图象数据的压缩和存储及图象特征的抽取等。

例1. 手写数字的识别

A small white box containing handwritten digits 1 through 9. The digits are written in a cursive, handwritten style. The digits are arranged in three rows: the first row contains 1, 2, 3, 4, 5, 6; the second row contains 7, 8, 9, 3, 4; the third row contains 1, 0, 0, 8, 7, 1.

由于手写数字变化很大，有传统的统计模式识别或句法识别很难得到

高的识别率，BP网络可通过对样本的学习得到较高的学习率。为了克服字体大小不同，我们选取这些数字的一些特征值作为网络输入。（可提取）特征如：

1, 2, 3, 7 : 具有两个端点；

0, 6, 8, 9: 具有圈； 2: 两个端点前后；

2.2 非线性连续变换单元组成的网络

对于一个样本，若具有那个特征，所对应的特征输入单元取值为1，否则为0。我们可选择34个特征，即输入单元个数为34。输出可取10个单元，即1个输出单元对应一个数字（该单元输出为1，其它为0）。如果选取200个人所写的1000个样本进行学习，使用三层BP网络，隐层单元数 n_1 应如何选择呢？

根据前面的经验公式，可得到下面结果：

$$\sum_{i=0}^n \binom{n_1}{i} > 1000 \Rightarrow \min n_1 = 10$$

2.2 非线性连续变换单元组成的网络

$$n_1 = \sqrt{m+n} + a = \sqrt{44} + a = 8 \sim 17$$

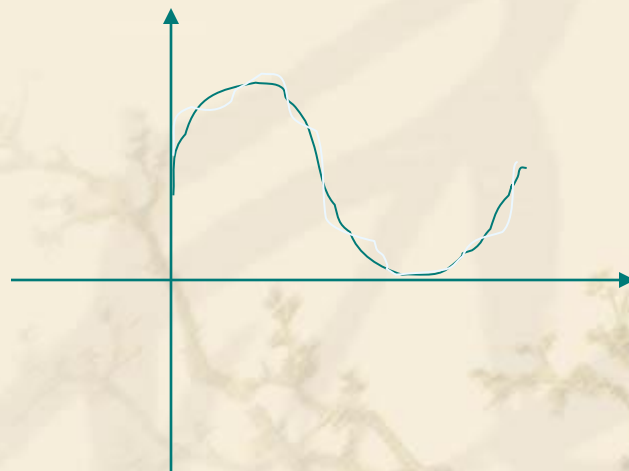
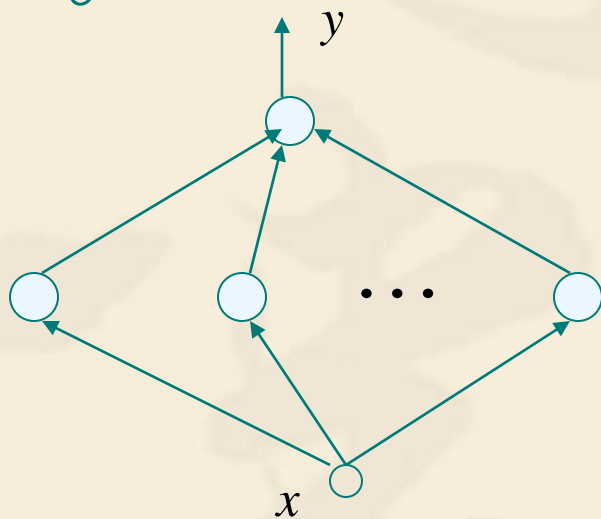
$$n_1 = \log_2 34 \approx 6$$

在实际中，我们选择 $n_1 = 14$ 。通过对1000个样本的学习所得到的网络对6000个手写数字的正确识别率达到95%。

例2. 非线性曲线的拟合。 在控制中往往希望产生一些非线性的输出输入关系。例如，已知一个机械臂取物的轨迹，根据这个轨迹可计算出机械臂关节的角度 θ_1 和 θ_2 （两个关节），按照机械臂的 θ 要求应该反演计算出驱动马达的力或频率这是一个相当复杂的计算问题。但我们可

2.2 非线性连续变换单元组成的网络

采用BP网络对一些样本的学习得到这些非线性曲线的拟合，根本无须知道机械臂的动力学模型。在一维情况下，就是拟合 $y = g(x)$ ，其中 x 表示 θ 角， y 为所对应的马达驱动力。在某些位置，我们容易得到这些对应值，因此可以得到足够的样本。



2.2 非线性连续变换单元组成的网络

例3. 数据压缩

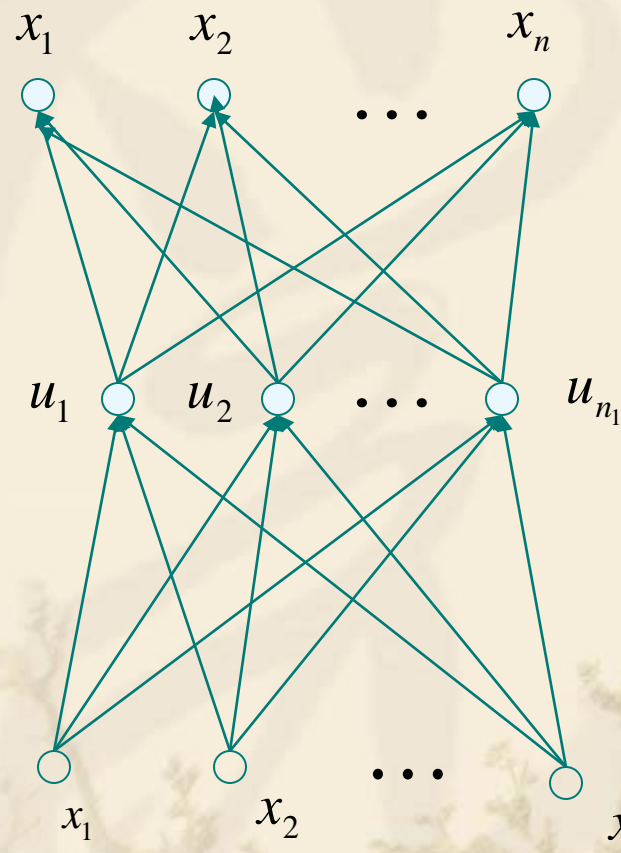
$$y = x, \quad n = m$$

$$y = F(u), \quad u = [u_1, u_2, \dots, u_{n_1}]^T, \quad n_1 < n$$

$$u = G(x) \leftrightarrow \text{对 } x \text{ 的编码}$$

$$n_1 / n \text{ --- 压缩率 } (n_1 = \log_2 n)$$

BP网络相当于一个编码、
解码器， n_1 越小，压缩
率越小，但太小可能达
不到唯一译码的要求。



2.2 非线性连续变换单元组成的网络

- 作业：1. 推导k层前馈网络的BP算法，并且考虑跨层连接的权值。
2. 采用2-2-1结构的前馈网络通过BP算法求解XOR问题，其中逼近精度 $\varepsilon = 0.001$ 。
3. 采用2-m-1结构的前馈网络通过BP算法来逼近定义于 $[0,1]^2$ 连续函数 $y = f(x) = 1/(1 + \sqrt{x_1^2 + x_2^2})$ ，其中逼近精度 $\varepsilon = 0.01$ 。请按均匀格点选择10000个样本点，随机选取5000个作为训练样本，且剩余的5000个作检测样本。根据该学习问题，可选取三种不同的m值，并观察所得网络在检测样本上的误差变化。