神经网络 Hopfield网络 学习问题

计算机科学 1994 Vol. 21 №. 5

30-34

非对称型 Hopfield 神经网络的学习问题*)

吴福朝 张 铃
(安徽大学计算机系 合肥 230039)

TP18

摘 要 In this paper, the learning problem of non-symetrical Hopfield neural networks is turned into solving unequalites and planning problem. Then famous ellipsoid method and planning method in mathematics have been applied to satisfactorily solve it.

关键词 Neural network、Ellipsoid method, Karmarkar method, Learing algorithm.

1. 引 言

联想记忆是人工神经元网络的重要功能之一，Hopfield 网络是一种重要的应用于联想记忆型网络。为了实现记忆功能，我们总希望通过训练使样本成为网络的稳定状态。然而，Hopfield 网络利用 Hebb 规则训练，求得的连接权矩阵 W：

$$w_{ij}=\begin{cases}\sum_{s=1}^{M} x_i^s x_j^s, & i\neq j\\ 0, & i=j\end{cases}$$

进行联想记忆时，稳定状态出现的规律并非如此简单，主要表现在[1]：

(a)给定一组样本，经过学习(训练)，网络将有一组稳定状态，但是给定的样本可能不是稳定状态。即使给定的样本均成为稳定状态、网络有可能出现很多其他多余的稳定状态(称之为伪稳定状态)，这些伪稳定状态会极大地损害联想记忆功能。

通过其他方式进行学习的任何对称型(即权矩阵为对称阵)Hopfield 网络，也会出现上述问题。因此有必要引进非对称型 Hopfield 网络[2]。文[2]的研究表明，非对称型 Hopfield 网络，较之对称型具有良好的记忆联想功能，但是它的结构却要复杂得多，这主要表现在：

(b)非对称型网络的稳定状态(吸引子)、有可能不是网络能量函数的极小点；而对称型网络在 $w_{ii}=0, \forall i$ 的情况下，所有稳定状态均为能量函数的极小点。

本文主要针对问题(a)和(b)，来讨论非对称网络的学习问题。即给定样本组：

$$\mathscr{X}^*=\{x^1,x^2,\cdots,x^N\}\subset\{-1,1\}^N$$

其中 N 为自然数，是训练样本 x^α 的维数，$1\leqslant\alpha\leqslant M$，求连接权矩阵 W^* 使之满足条件：

(Ⅰ)$\mathscr{X}^*$ 为相应网络的一组稳定状态；

(Ⅱ)$\mathscr{X}^*$ 中每一样本是相应网络能量函数的极小点；

(Ⅲ)相应网络的伪稳定状态要尽可能的少；

为了使相应网络有良好的联想功能，还要求满足条件：

(Ⅳ)$\mathscr{X}^*$ 中每一点的吸引半径>1。

无论是对称型还是非对称型，网络状态演化运行规律通常有同步运行和异步运行两种方式。

同步运行时，在每一单位时间中，网络的所有神

*)国家基础研究项目攀登计划基金资助。吴福朝 副教授。张铃 教授。

参考文献

[1]Zmoeeman, E. C., The Topology of the Brain and Visual Percaption. In Topology of 3—manifolds and Related Topics, Prentice-Hall, INC., Englewood Cliffs N. J., 1962

[2]陈霖，格式塔和容限空间，心理学报 1984，3

[3]杨行峻等，人工神经网络，高等教育出版社，1992，9

[4]王汝笠等，第六代计算机——人工神经网络计算机，科学技术文献出版社，1992，12

[5]Mao Guojun., Lattice-valued Logic, Intelligent Mathematics and Their Applications. Proc. IEEE ISMVL—89, 1989

[6]夏道行等，线性拓扑空间引论，上海科学技术出版社，1986，11

计算机科学 2

经元的状态同时进行调整，调整规则为：

$$x_i(t+1)=\mathrm{sgn}(\sum_{j=1}^{N}w_{ij}x_j(t)-\theta_i),\forall i。$$

异步运行时，以离散时间为变量，在每一个单位时间内，只对某个神经元 i 的状态进行调整，而其他状态保持不变。被调整的神经元可随机地选定，也可以按某规则选定。调整规则为：

$$\begin{cases}x_i(t+1)=\mathrm{sgn}(\sum_{j=1}^{N}w_{ij}x_j(t)-\theta_i)\\ x_j(t+1)=x_j(t),j\neq i\end{cases}$$

其中：$\mathrm{sgn}(s)=\begin{cases}1,s\geqslant 1\\ -1,s<0\end{cases}$是硬限幅函数，$\theta_i$ 为神经元 i 的阈值。

本文只讨论异步情况，同步可作类似讨论；并假定：$\theta_i=0,\forall i$。这不失一般性，因为任何网络可添加一个神经元使之 $\theta_i=0$；各种神经元具有硬限幅函数特性。

2 非对称型 Hopfield 网络的基本定理

本节一般地讨论非对称型 Hopfield 网络的基本性质，它们是解决本文所提出问题的根据。

令

$$x=\begin{pmatrix}x_1\\x_2\\ \vdots\\x_N\end{pmatrix},x(i)=\begin{pmatrix}x_ix_1\\x_ix_2\\ \vdots\\x_ix_N\end{pmatrix},i=1,2,\cdots,N.$$

$$W=\begin{pmatrix}w_1\\w_2\\ \vdots\\w_N\end{pmatrix}$$，w_i 为 W 第 i 行向量。

x 为稳定状态$\Leftrightarrow x=\mathrm{sgn}Wx\Leftrightarrow\forall i,x_i=\mathrm{sgn}(w_i,x)$

$$\Leftrightarrow\forall i,\begin{cases}(w_i,x(i))\geqslant 0,x_i=1\\(w_i,x(i))>0,x_i=-1\end{cases}$$

其中，$(\cdot,\cdot)$表示内积。

由于 W 的元素为有理数及 x(i) 的元素为整数，所以对充分小正数 ε，有：

$$x=\begin{pmatrix}x_1\\x_2\\ \vdots\\x_N\end{pmatrix}$$为网络的稳定状态$\Leftrightarrow$

$$\forall i,\begin{cases}(w_i,x(i))\geqslant 0,x_i=1\\(w_i,x(i))\geqslant\varepsilon,x_i=-1\end{cases}$$

令

$$X_x=(x(1),x(2),\cdots,x(N))$$

$$=\begin{pmatrix}x_1x_1 & x_2x_1 & \cdots & x_Nx_1\\ x_1x_2 & x_2x_2 & \cdots & x_Nx_2\\ \vdots & \vdots & & \vdots\\ x_1x_N & x_2x_N & \cdots & x_Nx_N\end{pmatrix}$$

$$b_x=\begin{pmatrix}b_1\\b_2\\ \vdots\\b_N\end{pmatrix}$$，当 $x_i=1$时，$b_i=0$；当 $x_i=-1$时，$b_i=\varepsilon$

这样，前述结果可以写成下面矩阵形式：

定理1 $x=\begin{pmatrix}x_1\\x_2\\ \vdots\\x_N\end{pmatrix}$为稳定状态$\Leftrightarrow(W,X_x)\geqslant b_x$。其中 $(W,X_x)=((w_1^T,x(1)),(w_2^T,x(2)),\cdots(w_N^T x(N)))^T$。

令 $x\in\{-1,1\}^N$，若 $d_H(x,y)=1$（d_H 表示海明距离），则称 y 为 x 的一个邻点。x 的任一邻点 y 必可表示为：$y=x+\delta_ie_i,1\leqslant i\leqslant N$。其中 e_i 为第 i 个元素为1的 N 维单位向量，$\delta_i=-2\mathrm{sgn}(x_i)$。

x 的 N 个邻点构成的集合表示为：

$$\{x+\delta_ie_i|1\leqslant i\leqslant N\}。$$

非对称型 Hopfield 网络的能量函数仍定义为

$$E(x)=-\frac{1}{2}x^TWx=-\frac{1}{2}(Wx,x)$$

若$\forall i,E(x)<E(x+\delta_ie_i)$，则称 x 为 E 的极小点。

下面分析 E 的极小点的特征：

$$\begin{aligned}\triangle E_i&=E(x+\delta_ie_i)-E(x)\\&=-\frac{1}{2}((W(x+\delta_ie_i),(x+\delta_ie_i))-(Wx,x))\\&=-\frac{1}{2}\delta_i((W+W^T)x,e_i)+\delta_i(We_i,e_i))\\&=-\frac{1}{2}\delta_i(\sum_{j=1}^{N}(w_{ij}+w_{ji})x_j+\delta_iw_{ii})\end{aligned}$$

(i) 当 $x_i=1$时，$\delta_i=-2\mathrm{sgn}(x_i)=-2$. 于是可推得：

$$\triangle E_i>0\Leftrightarrow(w_i^T+\bar{w}_i,x(i))-2w_{ii}>0$$

其中 $\bar{w}_i$ 是 W 的第 i 列向量。

(ii) 当 $x_i=-1$时，$\delta_i=2$. 于是可推得：

$$\triangle E_i>0\Leftrightarrow(w_i^T+\bar{w}_i,x(i))-2w_{ii}>0$$

综合(i)与(ii)，我们有

$$\triangle E_i>0\Leftrightarrow(w_i^T+\bar{w}_i,x(i))-2w_{ii}>0$$

令

$$d=\begin{pmatrix}w_{11}\\w_{22}\\ \vdots\\w_{NN}\end{pmatrix}$$为 W 的对角向量，$\varepsilon=\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\\ \vdots\\ \varepsilon_N\end{pmatrix}$为充分小的正向量，则根据 E 的极小点定义及上述讨论，有：

定理2 x 为 E 的极小点$\Leftrightarrow(W+W^T,X_x)-2d\geqslant\varepsilon$。特别地，在 $w_{ii}\leqslant 0,\forall i$ 的情况下，$(W+W^T,X_x)\geqslant\varepsilon\Rightarrow x$ 为 E 的极小点。

根据定理1和2立即有：

推论1 x为稳定状态且为E的极小点的充要条件为：

$$\begin{cases}(W,X_x)\geq b_x\\(W+W^T,X_x)-2d\geq \epsilon_x\end{cases}$$

下面分析稳定状态的邻点性质。

当 $w_{ii}=0,\forall i$，x为稳定状态时，任一邻点 $y=x+\delta_i e_i$ 均不是稳定状态，且对任一 $x+\delta_i e_i$ 可进行一步调整神经元i使 $x+\delta_i e_i$ 收敛于x。

事实上，由 $x_i=-y_i, x_j=y_j(j\neq i)$ 知

$$\mathrm{sgn}(\sum_{j\neq i}w_{ij}y_j)=\mathrm{sgn}(\sum_{j\neq i}w_{ij}x_j)=x_i=-y_i\neq y_i$$

当 $\exists\, i, w_{ii}\neq 0$，x为稳定状态时，令 $y=x+\delta_i e_i$ 即 $x_i=-y_i, x_j=y_j(j\neq i)$，由于我们只考虑异步运行网络，因此只须调整第i个神经元的状态：

$$\mathrm{sgn}(\sum_{j=1}^{N}w_{ij}y_j)=\mathrm{sgn}(\sum_{j\neq i}w_{ij}x_j+w_{ii}y_i)$$

$$=\mathrm{sgn}(\sum_{j\neq i}w_{ij}x_j-w_{ii}x_i)=\mathrm{sgn}(\sum_{j=1}^{N}w_{ij}x_j-2w_{ii}x_i)$$

所以

$$y=x+\delta_i e_i\text{ 不是稳定状态}\Leftrightarrow \mathrm{sgn}(\sum_{j=1}^{N}w_{ij}x_j-2w_{ii}x_i)=x_i$$

$$\Leftrightarrow \sum_{j=1}^{N}w_{ij}x_ix_j-2w_{ii}\begin{cases}\geq 0, x_i=1\\>0, x_i=-1\end{cases}$$

于是 $\forall i, x+\delta_i e_i$ 不是稳定状态 $\Leftrightarrow (W,X_x)-2d\geq c_x$，其中 c_x 为非负向量。

同时，还可以看出，若 $y=x+\delta_i e_i$ 不是稳定状态时，则可调整第i个神经元使y收敛于x。

总结以上讨论，我们有

定理3 令x为稳定状态，则 $\forall i, x+\delta_i e_i$ 不是稳定状态 $\Leftrightarrow (W,X_x)-2d\geq c_x\Leftrightarrow$ x的吸引半径>1。特别地，在 $w_{ii}\leq 0,\forall i$ 的情况下，$\forall i, x+\delta_i e_i$ 不是稳定状态且x的吸引半径>1。

3 满足各种条件的学习算法

本节给出求满足第1节中提出的各种条件的 W^* 的学习算法。

$\mathscr{X}=\{x^1,x^2,\cdots,x^M\}$ 为给定的学习样本组，

$$x^\alpha=\begin{pmatrix}x_1^\alpha\\x_2^\alpha\\\vdots\\x_N^\alpha\end{pmatrix},\alpha=1,2\cdots,M$$

3.1 求解满足条件(Ⅰ)和(Ⅱ)的 W^*

由第2节推论1，存在 W^* 使相应网络满足 $\mathscr{X}$ 为稳定状态组且为E的极小点的充要条件为不等式组：

$$\begin{cases}(W,X_{x^\alpha})\geq b_{x^\alpha}\\(W+W^T,X_{x^\alpha})-2d\geq \epsilon_{x^\alpha}\end{cases}\quad x^\alpha\in\mathscr{X}$$

关于W有解。

令w为W对应的N×N维向量，即

$$w=\begin{pmatrix}w_1^T\\w_2^T\\\vdots\\w_N^T\end{pmatrix}$$，其中 w_i^T 为W的第i行向量的转置。

$$S_k=\begin{pmatrix}x^1(k)^T\\x^2(k)^T\\\vdots\\x^M(k)^T\end{pmatrix}=\begin{pmatrix}x_k^1x_1^1 & x_k^1x_2^1 & \cdots & x_k^1x_N^1\\x_k^2x_1^2 & x_k^2x_2^2 & \cdots & x_k^2x_N^2\\\vdots\\x_k^Mx_1^M & x_k^Mx_2^M & \cdots & x_k^Mx_N^M\end{pmatrix},$$

$$1\leq k\leq N$$

$$S=\begin{pmatrix}S_1 & 0 & \cdots & 0\\0 & S_2 & \cdots & 0\\\vdots & \vdots & & \vdots\\0 & 0 & \cdots & S_N\end{pmatrix}$$

$$b_k=\begin{pmatrix}b_k(x^1)\\b_k(x^2)\\\vdots\\b_k(x^M)\end{pmatrix},k=1,2,\cdots,N$$

其中 $b_k(x^\alpha)$ 为 b_{x^α} 的第k分量，$\alpha=1,2,\cdots,M$

$$b=\begin{pmatrix}b_1\\b_2\\\vdots\\b_N\end{pmatrix}$$

可以看出不等式组 $(W,X_{x^\alpha})\geq b_{x^\alpha}, x^\alpha\in\mathscr{X}$ 变为下述矩阵形式：$Sw\geq b$。

令

$$H(r,k)=\begin{array}{c}\begin{pmatrix}0 & \cdots & 0 & x_r^1x_k^1 & 0 & \cdots & 0\\0 & \cdots & 0 & x_r^2x_k^2 & 0 & \cdots & 0\\\vdots & & \vdots & & \vdots & & \vdots\\0 & \cdots & 0 & x_r^Mx_k^M & 0 & \cdots & 0\end{pmatrix},\\ \text{第r列}\end{array}$$

$$r,k=1,2\cdots,N$$

例如：

$$H(1,2)=\begin{pmatrix}x_1^1x_2^1 & 0 & \cdots & 0\\x_1^2x_2^2 & 0 & \cdots & 0\\\vdots & \vdots & & \vdots\\x_1^Mx_2^M & 0 & \cdots & 0\end{pmatrix}$$

$$H=\begin{pmatrix}H(1,1) & H(1,2) & \cdots & H(1,N)\\H(2,1) & H(2,2) & \cdots & H(2,N)\\\vdots & \vdots & & \vdots\\H(N,1) & H(N,2) & \cdots & H(N,N)\end{pmatrix}$$

$$D=\begin{pmatrix}-2E_{11} & 0 & \cdots & 0\\0 & -2E_{11} & \cdots & 0\\\vdots & \vdots & & \vdots\\0 & 0 & \cdots & -2E_{11}\end{pmatrix},$$

$$E_{11}=\begin{pmatrix}1 & 0 & \cdots & 0\\0 & 0 & \cdots & 0\\\vdots & \vdots & & \vdots\\0 & 0 & \cdots & 0\end{pmatrix}_{M\times N}$$

ϵ 与b类似，是由 $\epsilon_{x^\alpha}, x^\alpha\in\mathscr{X}$ 导出的M×N维向量。

可以看出，不等式组：

$(W+W^T, X_{x^s})-2d\geq \epsilon_{x^s}, x^s\in\mathscr{X}^-$

变为下述矩阵形式：

$(S+H+D)w\geq\epsilon$。

于是，存在 W^* 使每一 $x^s\in\mathscr{X}$ 为稳定状态且为 E 的极小点的充要条件为不等式组

$$\begin{cases} Sw\geq b \\ (S+H+D)w\geq\epsilon \end{cases} \quad (1)$$

有解 w^*，且 w^* 相应的矩阵即为 W^*。

我们可以利用下面的椭球算法[3]解不等式组(1)。

椭球算法。

初始：令

(a) $i=0$

(b) $K:=2N^2((2N^2+1)\cdot L-N^4)$，L 为(1)的规模；

(c) $A_0:=R^2I, R=N2^{L-N^4}; a_0=0$；

$E_0:=E(A_0,a_0)=\{x:(x-a_0)^TA_0^{-1}(x-a_0)\leq 1\}$ 是初始椭球。

迭代：

(d)若 $i=K$，停止((1)式无解)；

(e)若 a_i 满足(1)式，停止(找到(1)式的一个解 a_i)；

(f)若 a_i 不满足(1)，则不等式组(1)存在某个不等式例如 $Sw\geq\gamma$ 使 $Sa_i<\gamma$；

令

(g) $b:=\frac{1}{\sqrt{SA_iS^T}}A_iS^T \quad a_{i+1}:=a^i-\frac{b^T}{N^2+1}$

$$A_{i+1}:=\frac{N^4}{N^4-1}(A_i-\frac{2}{N^2+1}b^Tb)$$

返回(d)。

椭球算法的时间复杂性为 $O(N^8)$。如果只求满足条件(Ⅰ)的权矩阵 W^*，利用椭球算法，其复要杂性可降低为 $O(N^4\log N)$[4]。

3.2求解满足条件(Ⅰ)、(Ⅱ)，和(Ⅳ)的 W^*。

由推论1和定理3，存在 W^* 满足条件(Ⅰ)、(Ⅱ)和(Ⅳ)的充要条件为不等式组：

$$\begin{cases} (W,X_{x^s})\geq b_{x^s} \\ (W+W^T,X_{x^s})-2d\geq\epsilon_{x^s} \quad x^s\in\mathscr{X}^- \\ (W,X_{x^s})-2d\geq C_{x^s} \end{cases}$$

关于 W 有解。

用3.1节中的类似方法，上述不等式组有解等价于下述不等式组：$\begin{cases} Sw\geq b \\ (S+H+D)w\geq\epsilon \\ (S+D)w\geq c \end{cases} \quad (2)$

关于 w 有解 w^*，且 w^* 的相应矩阵即为 W^*。

同样，我们可以利用椭球算法求解不等式组(2)，从而求得满足条件的连接权矩阵 W^*。

注：由定理2和定理3的后半部分，若 w^* 为不等式组：

$$\begin{cases} Sw\geq b \\ (S+H)w\geq\epsilon \end{cases}$$

的解且 $w_{ii}^*\leq 0, 1\leq i\leq N$，则 w^* 也必为(2)的解。因此，通常我们可以由不等式组

$$\begin{cases} Sw\geq b \\ (S+H)w\geq\epsilon \\ w_{ii}\leq 0, 1\leq i\leq N \end{cases}$$

的解 w^* 来确定满足条件(Ⅰ)，(Ⅱ)，(Ⅳ)的连接权矩阵 W^*，这可降低计算量。但上述不等式组无解时，我们不能断定原问题无解。

3.3求解满足条件(Ⅰ)与(Ⅱ)的连接权矩阵 W^*

x 不是网络的稳定状态⇔存在 i 使 $(w_i^T, x(i))<0$

令 $p_i=\{x\mid(w_i^T,x(i))<0\}, i=1,2,\cdots,N$；其中 $\|p_i\|$ 是 p_i 中元素的个数。

求使 $\mathscr{X}^-$ 为稳定状态组且伪稳定状态最少的连接权矩阵 W^*，等价于求解 N 个整数规划：

$$\begin{cases} \max\|p_i\| \\ \text{subject to}(w_i^T, x^s(i))\geq b_i(x^s), x^s\in\mathscr{X}^- \end{cases}$$

其中：$i=1,2,\cdots,N$。当 $x_i^s=1$ 时，$b_i(x^s)=0$；当 $x_i^s=-1$ 时，$b_i(x^s)$ 为小正数。

显然，这是 NP－问题。下面给出一种近似解法。从下式：

$$(w_i^T, \sum_{x\in\{-1,1\}^N}x(i))=\sum_{x\in\{-1,1\}^N}(w_i^T,x(i))$$

可以看出：$(w_i^T, \sum_{x\in\{-1,1\}^N}x(i))$ 的值越小，$\|p_i\|$ 就可能越大，而

$$2^Ne_i=\sum_{x\in\{1,-1\}^N}x(i)$$

$$(w_i^T, \sum_{x\in\{1,-1\}^N}x(i))=2^N(w_i^T,e_i)=2^Nw_{ii}$$

因此，我们将下述线性规划 LP(i)：

$$\begin{cases} \min w_{ii} \\ \text{subject to}(w_i^T, x^s(i))\geq b_i(x^s), x^s\in\mathscr{X}^- \end{cases}$$

$i=1,2,\cdots,N$

的解，作为前述整数规划的近似解。

由于 $\mathscr{X}$ 中每一状态为稳定状态⇔$Sw\geq b$，所以，求解满足条件(Ⅰ)，(Ⅱ)的连接权矩阵 W^*，可通过下述线性规划：

$$\begin{cases} \min\sum_{i=1}^{N}w_{ii} \\ \text{subject to } Sw\geq b \end{cases} \quad (3)$$

的解 w^* 来确定。

(3)式可用 Karmarkar 方法[5,6]在多项式时间内求解。

3.4 求解满足条件(Ⅰ)—(Ⅳ)的连接权矩阵 W^*.

由3.2节和3.3节的讨论,求满足(Ⅰ)—(Ⅳ)的权矩阵 W^*,等价于求下述线性规划:

$$\begin{cases} \min \sum_{i=1}^{N} w_{ii} \\ \text{subject to} \begin{cases} Sw \geqslant b \\ (S+H+D)w \geqslant \epsilon \\ (S+D)w \geqslant c \end{cases} \end{cases} \quad (4)$$

的解 w^*.

当不等式组:

$$\begin{cases} Sw \geqslant b \\ (S+H)w \geqslant \epsilon \\ w_{ii} \leqslant 0 \end{cases}$$

有解时,根据3.3节法,也可通过求解下述线性规划:

$$\begin{cases} \min \sum_{i=1}^{N} w_{ii} \\ \text{subject to} \begin{cases} Sw \geqslant b \\ (S+H)w \geqslant \epsilon \end{cases} \end{cases} \quad (5)$$

来确定 W^*。

参考文献

[1]协行俊,郑君里,人工神经元网络,高教出版社(1990)

[2]张承福,王心强等,几种联想记忆神经网络模型的分析,人工智能与模式识别,3卷1期(1990),p14-21

[3]Martin Grotschel et al., Geometric Algorithms and Combinational Optimizatiom, Springer Verlag 1990.

[4]Bo Zhang, Ling Zhang, Fuchao Wu, Programming Based Learning Algorithm of Neural Networks with Self-feedback Connections, IEEE Trans, Pattern Anal. Machine Intell. To appear.

[5]N. Karmarkar, A New Polynomial-time Algorithm for Linear Programming, Combinatorica 4(1984), 373-395

[6]G. Strang, Introduction to Applied Mathematics, Wellesley-Cambridge Prass(1986)

(上接第55页)

必要的后处理(例如排序、成组、连接)。另外,UniSQL/M支持局部数据库之上的"分布式事务管理",这意味着一个UniSQL/M事务中发出的所有更新即使会产生对多个局部数据库的更新,但也能同时被提交或夭折。

当今,RDB卖主提供不同复杂程度的信关。有些信关允许SQL查询被传递到层次数据库系统(即IMS)或者文件系统(如DEC的RMS)。有些信关当前正被提升为能接受查询和更新、甚至支持局部数据库上的分布式事务管理。但是,这些信关中还没有一个能将SQL查询传递到OODB的;也很少有开发这样的信关的要求。

UniSQL/M在三个主要方面不同于RDB卖主和OODB卖主当前提供的信关。

——UniSQL/M是一个象样的数据库系统,而不仅仅是信关,它支持查询、更新、授权、以及在全局数据库(关于定义在局部数据库的表和类上的视图说明,还有局部数据库的表和类的信息目录)上的事务管理,当前的大多数信关不接受更新。

——UniSQL/M为处理单个UniSQL/M事务要连到局部数据库上,并协调多个局部数据库的查询和更新;具体来说,它支持局部数据库上的分布式事务管理。当前大多数信关只是将请求传递给一个局部数据库,或者当它们支持多个局部数据库时,也不允许在单个事务中同时对多个局部数据库进行更新。

——UniSQL/M还提供了一个比当前任何信关都更强有力的设施。尽管不够充分(由于理论限制),UniSQL/M还是将局部RDB扩充成了UniSQL/X;也就是说,UniSQL/M能对局部关系数据库中所检索到的元组增加对象标识并允许用户为其加上方法,将它们转换成对象。用这个方法,UniSQL/M使得UniSQL/X中提供的主要的面向对象设施对局部RDB也可用,具体包括SQL/X的路径查询、方法、以及UniSQL/M内存对象的工作空间管理。

UniSQL/M至少在三个不同的环境下是可用的。首先,可以用于允许UniSQL/X和RDB的共存。其次,可以用于将一组RDB(或一组UniSQL/X)变成一个分布式数据库系统。最后,当与单个RDB打交道时,它可以作为RDB模型上的对象管理层,把RDB变成UniSQL/X。(全文完)

[张成洪译自《Proceedings of the 19th VLDB Conference》, Dublin, Ireland 1993, 施伯乐校]