

用神经网络求解时延、功耗和连线 三重驱动的布局问题*

胡卫明

李翠超 朱育松 严晓浪

(北京大学计算机科学技术研究所 北京 100871) (杭州电子工业学院 CAD 所 杭州 310037)

摘要 本文应用 Kohonen 自组织神经网络求解时延、功耗和连线三重驱动的门阵列布局问题。算法用自组织学习算法和分配算法确定关键单元的位置,用迭代改善的方法确定非关键单元的位置,从而获得关键线网最短、散热大的单元离得尽可能远并且单元连线总长尽可能短的布局。本文还介绍了面向线网和功耗的样本矢量的概念,与面向单元的样本矢量相比,面向线网和功耗的样本矢量不仅可以直接处理多端线网,而且能够描述时延信息和热信息。实验表明这是一种有效的方法。

CCACC: 7410D, 5120

1 引言

随着集成电路工艺的迅猛发展,人们对集成电路的性能要求日益提高,研究人工神经网络在性能驱动的集成电路布局中的应用,是一个有现实意义的研究课题。

时延、功耗和连线是性能驱动的集成电路布局中应该考虑的问题。本文应用 Kohonen 自组织神经网络求解时延、功耗和单元总连线长度三重驱动的门阵列布局问题。算法分三步:

第一步:用自组织学习算法确定关键单元的基本位置,其中,关键路径上的线网称为关键线网,关键线网所联单元和热源单元称为关键单元,其它的单元均为非关键单元。

第二步:用分配算法处理关键单元映射的重叠现象。

第三步:用迭代改善的方法确定非关键单元的位置。在迭代改善过程中,以连线总长度最短为优化目标。这样就可以在保证关键线网连线最短、热源单元离得尽可能远的前提下,使得单元间连线总长尽可能短。

2 神经网络模型

Kohonen 自组织神经网络是由输入层和输出层组成的双层神经网络。输入层中的每一

* 本文研究得到国家自然科学基金和国家“九五”重点科技攻关项目资助

胡卫明 1968 年出生,计算机博士、博士后,主要从事 IC-CAD、人工神经网络和 GIS 的研究工作

严晓浪 1947 年出生,博士生导师,从事 IC-CAD 和设计自动化领域的教学与科研

1998-03-03 收到,1998-06-25 定稿

个神经元,通过权与输出层的每一个神经元相联 设 $X \in R^M$ 为输入模式向量, W 为权值矩阵, $Y \in R^N$ 为输出节点的匹配响应,在 t 时刻有 $Y = W^T(t) \oplus X(t)$. \oplus 代表一种运算,一般有两种形式: (1) 点积运算,即 $Y = W^T(t) \cdot X(t)$, 输入样本 X 与权的内积最大值为竞争优胜者; (2) Euclidean 距离运算,即 $Y = \|W(t) - X(t)\|_g$, 竞争得胜的神经元对应最小的输出值

在输出层中,竞争是这样进行的: 对于“赢”的那个神经元 c , 在其周围 N_c 的区域内神经元在不同程度上得到兴奋,而在 N_c 以外的神经元都被抑制,即:

$$dw_{ij}/dt = \begin{cases} \alpha(t)(x_i(t) - w_{ij}(t)) & \forall j \in N_c \\ 0 & \forall j \notin N_c \end{cases}$$

N_c 随着 t 的增大逐渐缩小, $\alpha(t)$ 随 t 的增大逐渐减小

网络的学习过程就是网络的连接权根据训练样本进行自适应、自组织的过程,经过一定次数的训练以后,网络能够把拓扑意义下相似的输入样本映射到相近的输出节点上 网络能够实现从输入到输出的非线性降维映射

3 样本矢量的建立

3.1 面向单元的样本矢量

面向单元的样本矢量^[2,3](文献[2]对间接连接单元的连接度只做一次修正,文献[3]对此做了改进,较好地解决了非直接连接单元的连接度问题)是以单元间的连接度作为它们之间相似性量度的,网络自组织的结果使得连接度大的单元被映射到布局平面上邻近的位置上 它有三个不足之处: (1) 不能直接表达时延信息 (2) 不能直接表达热信息 (3) 虽能直接处理两端线网,但对于多端线网在产生单元连接矩阵时,需把多端线网等效成适当连接权的两端线网 这种等效方法具有一定的失真性,不能完全真实反映多端线网的连接特征

3.2 面向线网和功耗的样本矢量

面向线网和功耗的样本矢量将时延信息和热信息描述在样本矢量的特征分量中,由关键线网特征分量和热特征分量组成: 以关键线网的加权值作为关键线网特征分量; 用单元的散热量作为热特征分量(时延和热的处理思想与文献[4]类似). 具有相同特征值的相似单元经自组织学习以后能够移到一起或因为都是热源单元而移得较远 显然处理的每一步用的都是关于电路的完整的、真实的连接信息,而不是分解的、变形的信息,从而能够直接处理多端线网

获取线网权值的一个方法是: 先对电路进行时延分析,通过时延方程、组合电路单元的内部时延和系统周期时间确定线网的权值 为了网络训练的方便,还要将关键线网的权值除以线网的最大权值,使得线网的权值小于等于 1 在许多场合,关键路径是由用户指定的,这时可以直接用用户指定的关键路径上的线网作为特征分量

芯片中的热来自各晶体管 对于 CMOS 电路,门在“开”状态和“关”状态下产生的热量是不一样的,需确定在任一时刻“开”状态门的数目和“关”状态门的数目的比例 对于 ECL 系统,典型的门功耗为 0.25mW/每门,而与门的状态和运行频率无关 输入样本矢量的热分量需要除以单元的最大散热量,使得每个样本的热分量小于等于 1

图 1 是一个简化的单元连接图,设线网 n_1 、 n_2 、 n_3 和 n_4 都为关键线网,单元 b 和 d 为热源单元,关键线网 n_4 为三端线网,则该单元连接图的输入样本如表 1 所示

表 1 图 1 的样本数据

单元	特征量				热特征
	关键线网 n_1	关键线网 n_2	关键线网 n_3	关键线网 n_4	
a	0.95	0	0	0.85	0
b	0	1	0	0.85	0.8
c	0.95	1	0.8	0	0
d	0	0	0.8	0.85	1

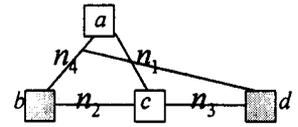


图 1 单元连接图

4 自组织学习算法

对应于一个有 K 个关键单元, $M - 1$ 个关键线网和布局平面上有 N 个单元放置位置 (即输出节点) 的门阵列布局实例, 网络需要 K 个训练样本点, M 个输入矢量特征分量 ($M - 1$ 个关键线网分量和一个热分量) 和 N 个输出神经元 其中, 一个关键单元对应一个样本; 一个单元放置位置对应一个输出神经元 网络的自组织学习过程为:

步骤 1: 随机给定网络初始权: $0 < w_{ij} < 1, i = 1, 2, \dots, M, j = 1, 2, \dots, N$

步骤 2: 从输入样本数据中随机选取一样本点 X , 计算所有输出节点的输出值:

$$y_j = \sum_{i=1}^M x_i w_{ij}, j = 1, 2, \dots, N$$

步骤 3: 选择输出值最大的输出节点 c 为竞争获胜神经元:

$$y_c = \max_j (y_j), j = 1, 2, \dots, N$$

步骤 4: 确定神经元 c 的领域 N_c

步骤 5: 更新 N_c 内所有输出神经元 j 的权值:

$$w_{ij} = w_{ij} + \alpha(t) \eta_j (x_i - w_{ij}), i = 1, 2, \dots, M - 1, j \in N_c$$

$$w_{Mj} = w_{Mj} + \alpha(t) \eta_j (x_M - w_{Mj}), i = M, j = c$$

$$w_{Mj} = w_{Mj} + \alpha(t) (1 - \eta_j) (x_M - w_{Mj}), i = M, j \in N_c, j \neq c^{[4]}$$

其中

$$\alpha(t) = A e^{-t/\tau} (\tau \text{ 为常数}) \text{ 或 } \alpha(t) = 0.9(1 - t/1000)$$

$i = M$ 为热特征分量

$$\eta_j = Y^d (1 < Y < 2), (d \text{ 为输出神经元 } c \text{ 和 } j \text{ 的 Manhattan 距离})$$

步骤 6: 转步骤 2

当所有样本输入一遍后, 满足 $\max_{i,j} \lfloor w_{ij}(t+1) - w_{ij}(t) \rfloor < \epsilon$ 或达到预先取定的迭代次数后, 学习结束; 否则进入下一轮学习

进一步说明如下:

(1) 输入矢量与输出神经元的权矢量的相似性用它们的点积值 (而非 Euclidean 距离) 表示

(2) 热特征分量 ($i = M$) 的学习规则必须特殊考虑, 它的学习方式与其它特征不同

(3) 输出神经元 c 在不同时间 t 时的邻域 $N_c(t)$ 由与神经元 c 间的距离小于等于 $k(t)$ 的神经元组成 $k(t)$ 反映了领域的大小, 是一个时间递减函数 初始时邻域可以取很大, 随着学习步数 t 的增大再线性缩小, 直到只剩下一个神经元

(4) 参数 τ 和 γ 越大, 网络收敛得越快, 但解的质量会受影响

(5) 收敛条件中的 ϵ 越小或选取的迭代次数越多, 解的质量越好, 但所需的计算时间越长

由于样本训练次数一般为 $O(K)$ 数量级, 每输入一个样本平均有 \log_2^N 个输出神经元的权值需要调节, 且每个输出神经元有 M 个权分量, 所以该算法的时间复杂性为 $O(KM \log_2^N)$. 另外, 实际样本训练次数还与初始状态有关, 不同的初始状态, 迭代到满足收敛条件, 所需的迭代步数是不完全一样的

5 分配算法

对于上述的自组织学习过程, 如果学习时间不够长(在电路规模较大的情况下, 要使网络收到最优解需要很长的学习时间), 那么多个关键单元会映射到同一个输出节点上, 即有多个样本在同一输出神经元上有最大输出值, 称之为单元重叠. 单元重叠出现的可能性主要取决于关键单元在单元集合中所占的比例, 关键单元越多, 越容易出现重叠. 但一般情况下, 关键单元占的比例并不大, 所以重叠现象并不严重, 而且由于关键单元比输出节点要少很多, 即使在出现重叠的情况下, 也很容易为重叠单元找到一个次优匹配的输出节点, 因此一般情况下分配后对布局结果的影响并不大. 我们设计了一个基于贪心思想的分配算法来处理这种情况. 算法描述如下:

第一步: 初始化关键单元集合 A 和输出节点集合 B , 即 $A = \{1, 2, \dots, K\}, B = \{1, 2, \dots, N\}$.

第二步: 相对于样本 i , 输出节点 j 的输出值用 y_{ij} 表示. 如果

$$y_{a,b} = \max_{i \in A, j \in B} (y_{ij}), a \in A, b \in B,$$

则将输出节点 b 分配给关键单元 a , 并且 $A = A - \{a\}, B = B - \{b\}$.

第三步: 如果集合 A 非空, 则转到第二步, 否则结束

6 非关键单元的定位

自组织学习算法和分配算法可以确定关键单元在布局平面上的位置. 对于门阵列版图模式, 可以用 Steinberg 法、对交换法、邻域交换法、力向量成对交换法、力向量松弛法、广义力向量松弛法等迭代改善方法来减少单元的总连线长度, 以确定非关键单元在布局平面上的位置. 由于此时已与关键线网、热源单元无关, 所以也可以用传统的提高布通率的算法来确定非关键单元的位置.

7 实验结果

本算法已用 Borland C++ 编程, 在 486/66 计算机上实现. 两类布局实例被用来验证算法的有效性: 第一类为规则的筛网结构电路, 并在电路上指定一条或多条关键路径. 这类例子的最佳布局构型是已知的, 比较容易判断一个布局结果与最佳布局有多大偏差. 另一类例子由计算机随机产生, 它更具一般性. 表 2 给了测试用的 4 个电路的有关信息, 其中例 1 和

例 2 为第一类电路, 例 3 和例 4 为第二类例子.

表 2 实验电路的有关信息

例号	单元数	线网数	关键单元数	关键线网数	2 端线网数	3 端线网数	4 端线网数
1	25	40	5	4	40	0	0
2	25	40	9	7	40	0	0
3	100	210	39	71	136	52	22
4	200	454	93	141	241	116	97

表 3 给出了例 2 的样本数据, 图 2 和表 4 是例 2 在不同时刻的执行情况和运行结果. 从结果可以看出, 关键线网的长度达到最短, 热源单元 1、4、7 和 9 分得较开.

表 3 例 2 的样本数据

单元号	特征量							散热
	n_1	n_2	n_3	n_4	n_5	n_6	n_7	
1	0.9	0	0	0	0	0	0	1
2	0.9	0.95	0	0	0	0	0	0
3	0	0.95	0.85	0.87	0	0	0	0
4	0	0	0.85	0	0	0	0	0.9
5	0	0	0	0.87	0	0	0	0
6	0	0	0	0	0.92	0	0	0
7	0	0	0	0	0	1	0	0.8
8	6	0	0	0	0.92	1	0.96	0
9	0	0	0	0	0	0	0.96	0.85

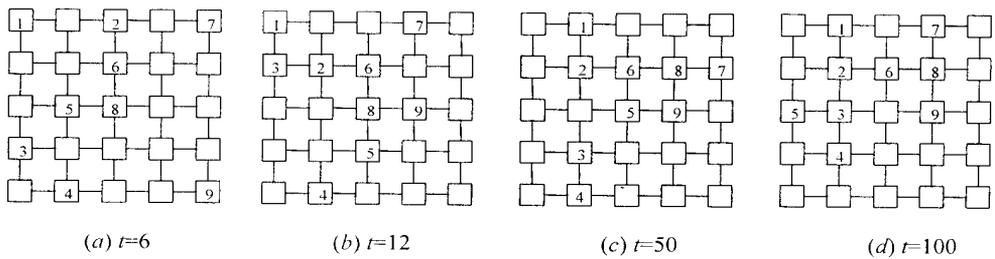


图 2 例 2 在不同时刻 t 的执行情况

表 4 例 2 在不同时刻的运行结果

测试量 (关键线网总长)	$t=0$	$t=6$	$t=12$	$t=25$	$t=50$	$t=100$
Euclidean 平方和	51	36	29	20	11	7
Manhattan 距离	29	20	16	12	9	7

表 5 比较了本算法与 HNN 算法的实验结果. 其中 HNN 算法是由文献[1]中的算法改进而来的: 文献[1]优化的是连线总长, HNN 算法改进为优化关键线网的长度. 本算法的运

行结果依赖于随机产生的网络初始权值,表5的例3、例4是5次运行的最佳结果(例1、例2的规模较小,每次运行均能达到最优解)。从表中可以看出,本算法不仅运行速度快于HNN算法,而且布局的质量也优于HNN算法

表5 实验结果

例号	关键线网 Euclidean 总长		关键线网 Manhattan 总长		运行时间/s	
	本算法	HNN 算法	本算法	HNN 算法	本算法	HNN 算法
1	4	4	4	4	0.05	0.07
2	7	9	7	10	0.12	0.53
3	112	125	139	164	8.40	23.9
4	236	248	274	297	90.3	264.0

8 结束语

本文应用 Kohonen 自组织神经网络求解以关键线网时延最小、散热大的单元相距尽可能远并且单元间连线总长度尽可能短为优化目标的门阵列布局问题。与文献[4]相比,本算法的特点是:是门阵列布局,而后者是任意单元布局;建立了面向线网的样本矢量能够直接处理多端线网的思想;用分配算法处理关键单元重叠现象;能优化单元间的连线

参 考 文 献

- [1] Sriram M. , Kang S.M. , Proc of ISCAS, 1990: 1664~ 1667.
- [2] Hemani A. , Postula A. , Neural Networks, 1990, 3(1): 377~ 383.
- [3] 沈涛,等,电子学报,1992, 20(10): 100~ 105.
- [4] Zhang C. X. , Proc of ISCAS, 1993: 2067~ 2070.
- [5] 焦李成,神经网络计算,西安:西安电子科技大学出版社,1995,第2章.
- [6] Unaltuna M. K. , Pitchumani V. , Proc of ISCAS, 1993: 2047~ 2050.
- [7] Gloria A. D. , Faraboschi P. , Olivieri M. , IEEE Trans CAD, 1994, 13(6): 694~ 701.

Neural Network Approach for Timing, Power Dissipation and Wire Connection Driven Placement

Hu Weiming

(Institute of Computer Science and Technology, Peking University, Beijing 100871)

Li Cuichao, Zhu Yulong, Yan Xiaolang

(CAD Center, Hangzhou Institute of Electronics Engineering, Hangzhou 310037)

Received 3 March 1998, revised manuscript received 25 June 1998

Abstract The Kohonen self-organizing neural network approach is applied to the timing, power dissipation, and wire connection driven placement of gate array. In the algorithm, the critical nets are minimized, the distance between heat source cells is maximized, and the total length of nets is minimized. The self-organizing learning approach and the assignment algorithm are used to fit the critical cells, and traditional iterative improvement method is used to fit the other cells. In addition, the concept of net and power dissipation oriented similarity vector is introduced. Compared with the cell oriented similarity vector, net and power dissipation oriented similarity vector can not only deal with the multi-terminal nets directly, but also describe the information about timing and heat. The experimental result shows that it is an effective method.

CCACC: 7410D, 5120