Hopfield 网的图灵等价性

孟祥武[1,2] 程 虎[1]

[1](中国科学院软件研究所 北京 100080)

[2](北京邮电大学计算机工程系 北京 100876)

摘要 本文给出了用 Hopfield 网计算部分递归函数的构造性证明. 由于部分递归函数与图灵机等价,故 Hopfield 网与图灵机等价.

关键词 Hopfield 网,图灵机,可计算性,部分递归函数,神经网络.

中图法分类号 TP18

60 年代,Minsky 证明反馈网络可接受正规文法,即相当于有限自动机.[1] 1991 年,Sun 等证明了二阶权神经网络的图灵(Turing)等价性.[2] 1995 年,刘晓鸿等证明了线性阈值单元神经网络的图灵等价性.[3] 本文给出了 Hopfield 网的图灵等价性证明,即 Hopfield 网是一通用计算模型,在计算能力上与图灵机等价. 刘晓鸿等用不带反馈的线性阈值单元神经网络构造了部分递归函数. 本文将刘晓鸿等的结果推广到反馈网上,用一种反馈神经网络,即 Hopfield 网来构造部分递归函数,揭示了 Hopfield 网的计算能力.

1 Hopfield 网与图灵机

Hopfield 网是一常用的神经网络模型[4],主要用于联想记忆和优化计算.

离散型 Hopfield 神经网络,有 n 个神经元,权矩阵为 $n\times n$ 阶实数矩阵. 每个神经元的状态为 1 或 -1. 神经元 j 在时间 k 的状态表示为 $X_k(j)$,整个网络在时间 k 的状态为:

$$X_k=[X_k(1),\ldots,X_k(n)]^T$$

Hopfield 网按以下方式随时间演化:

$$X_{k+1}=\mathrm{sgn}(WX_k-\theta),\ X\in\{-1,1\}^n,\ W\in R^{n\times n},\ \theta\in R^n,$$

$$\mathrm{sgn}(x)=\begin{cases}1, & 当\ x\geqslant 0;\\ -1, & 当\ x<0.\end{cases}$$

对任一离散型 Hopfield 网,给定不动点、极限环或迭代序列,可构造满足这些条件的网络. 当样本无矛盾时,可构造出相应的权矩阵.[5]

部分递归函数是由后继函数、常数函数和投影函数在复合、原始递归和无界搜索运算下所构成的封闭类. 以上 3 个函数为本原函数.

由于实数、复数均可表示为自然数或自然数序列,而非数值信息也可用自然数编码的方式加以处理. 部分递归函数可以通过适当的编码定义于自然数集上.[6]

本文尝试用 Hopfield 网去模拟部分递归函数.

定理 1. 对每个 Hopfield 网络,都存在一个计算该网络的图灵机.

证明:因 Hopfield 网的输入是有限的,该函数通过适当的编码可定义于自然数集上,而有限累加求和及大小比较均为原始递归函数,故单个 Hopfield 网可用图灵机计算. 同时,神经网络间的联接可用函数复合表示,故形成的网络也可由图灵机去计算. □

定理 2. 对部分递归函数类中任一函数,都存在一个计算该函数的 Hopfield 神经网络.

证明:部分递归函数是以本原函数为初始函数,使用 3 种运算构成的函数集. 下面就各种情况分别构造证明.

---

* 作者孟祥武,1966 年生,博士,讲师,主要研究领域为神经网络,进化算法. 程虎,1938 年生,研究员,博士导师,主要研究领域为语言编译,软件工程,人工智能和神经网络.

本文通讯联系人:孟祥武,北京 100876,北京邮电大学计算机工程系 135 信箱

本文 1996-11-20 收到原稿,1997-04-11 收到修改稿

(1)后继函数

$$S(x)\triangleq x+1 \quad x=b_kb_{k-1}\ldots b_1b_0(\text{二进制}) \quad b_i=\begin{cases}0, \\ 1\end{cases} \quad i=0,1,\ldots,k;$$

其中 b_i 表示第 i 位，c_i 表示第 i 位的进位位.

开始时将所有进位位设置为 0，$c_j=0$，$j=-1,0,1,\ldots,k$.

下面构造一 Hopfield 网，完成后继函数，即逐位加 1 功能.

初始状态(输入)				稳定状态(输出)			
c_{i-1}	b_i	c_i	奇偶校验位	c_{i-1}	b_i	c_i	奇偶校验位
0	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0
1	0	0	1	1	0	1	1
1	1	0	0	1	1	1	0

其中 $i=0,1,\ldots,k$.

这里 0 表示输入 -1，状态 0 表示状态 -1，1 表示输入 1，状态 1 表示状态 1.

对于每一位 b_i，可构造一 Hopfield 网，该网中有 4 个结点，即 c_{i-1},b_i,c_i，奇偶校验位，输入为初始状态值，即输入 -1 或 1，稳定状态即函数值.

构造满足上述条件的 Hopfield 网，相应的权矩阵可为 $\begin{bmatrix}4 & 1 & 1 & 2\\ 1 & 1 & 1 & -4\\ 2 & 2 & 1 & 2\\ 2 & 1 & 2 & 2\end{bmatrix}$. 其阈值均为 0，这里称该 Hopfield 网为 h0.

故后继函数可用图 1 的神经网络逐位计算本位和进位位而得到.

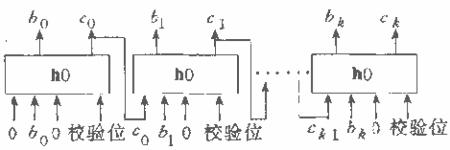

图1 后继函数神经网络

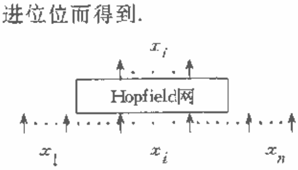

图2 投影函数神经网络

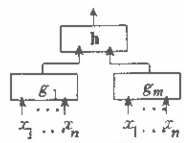

图3 复合函数神经网络

(2)常数函数

$$C(x)\triangleq m, m \text{ 为非负整数}. \quad m=b_kb_{k-1}\ldots b_1b_0(\text{二进制}) \quad b_i=\begin{cases}0, \\ 1\end{cases} \quad i=0,1,\ldots,k.$$

可构造一个有 $k+1$ 结点的 Hopfield 网，权矩阵为单位矩阵，此时 Hopfield 网的输入与输出一致，即实现了该常数函数.

(3)投影函数

$$P(x_1,x_2,\ldots,x_i,\ldots,x_n)\triangleq x_i$$

该 Hopfield 网的结点数即分量 x_i 的输入数目，其权矩阵为单位矩阵，只输出第 i 个变量(如图 2 所示).

(4)复合函数

$$f(x_1,\ldots,x_n)\triangleq h(g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n))$$

$g_1,\ldots,g_m,h$ 函数均为部分递归函数，且均用一 Hopfield 网构造实现(如图 3 所示).

(5)原始递归函数

$$f(0,x_2,\ldots,x_n)\triangleq g(x_2,\ldots,x_n)$$

$$f(x_1+1,x_2,\ldots,x_n)\triangleq h(x_1,f(x_1,\ldots,x_n),x_2,\ldots,x_n)$$

为了计算该函数，先构造选择函数

$$A(x_1,x_2,x_3,x_4)\triangleq\begin{cases}x_3, & x_1=x_2\\ x_4, & x_1\neq x_2\end{cases}$$

$$x_i=b_{ik}\ldots b_{i1}b_{i0}(\text{二进制}), \quad i=1,2,3,4$$

$$b_{ij}=\begin{cases}0, \\ 1\end{cases} \quad j=0,1,2,\ldots,k.$$

再用另外两个函数的复合来定义 A 函数.

相等判定　　$equal(x_1,x_2) \triangleq \begin{cases} 1, & x_1 = x_2 \\ 0, & x_1 \neq x_2 \end{cases}$

布尔选择　　$select(b,x_1,x_2) \triangleq \begin{cases} x_1, & b=1 \\ x_2, & b=0 \end{cases}$

构造一 Hopfield 网，判断输入值是否相等.

初始状态(输入)			稳定状态(输出)		
b_{1i}	b_{2i}	奇偶校验位	b_{1i}	b_{2i}	奇偶校验位
0	0	0	1	1	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	1	1	1

其中 $i=0,1,\ldots,k$;

这里 0 表示输入 -1，状态 0 表示状态 -1，1 表示输入 1，状态 1 表示状态 1.

构造满足上述条件的 Hopfield 网，相应的权矩阵可为 $\begin{bmatrix} 1 & 1 & -3 \\ 1 & 1 & -3 \\ 0 & 0 & 0 \end{bmatrix}$. 其阈值均为 0，这样的 Hopfield 网，b_{1i} 与 b_{2i} 相等输出为 1，否则为 0. 这里称该 Hopfield 网为 **h1**.

另构造一 Hopfield 网，判断输入值 b_{1i} 与 b_{2i} 是否均为 1.

初始状态(输入)			稳定状态(输出)		
b_{1i}	b_{2i}	奇偶校验位	b_{1i}	b_{2i}	奇偶校验位
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	1	1	1

其中 $i=0,1,\ldots,k$.

这里 0 表示输入 -1，状态 0 表示状态 -1，1 表示输入 1，状态 1 表示状态 1.

构造满足上述条件的 Hopfield 网，相应的权矩阵可为 $\begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$. 其阈值均为 0，该 Hopfield 网，$b_{1i}$ 与 b_{2i} 均为 1 输出为 1，否则为 0. 这里称该 Hopfield 网为 **h2**. 可利用 **h1** 和 **h2** 实现 equal 函数. 图 4 是实现 equal 函数的神经网络.

再构造一 Hopfield 网实现 select 函数.

初始状态(输入)				稳定状态(输出)			
b	b_{1i}	b_{2i}	奇偶校验位(关于 b,b_{1i})	b	b_{1i}	b_{2i}	奇偶校验位
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1

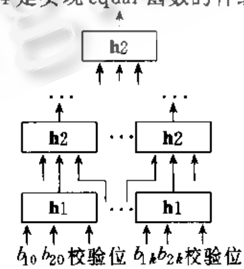

图4　equal函数神经网络

其中 $i=0,1,\ldots,k$.

这里 0 表示输入 -1，状态 0 表示状态 -1，1 表示输入 1，状态 1 表示状态 1.

构造满足上述条件的 Hopfield 网，相应的权矩阵可为 $\begin{bmatrix} 1 & 2 & 1 & -2 \\ 1 & 2 & 1 & -2 \\ 1 & 2 & 1 & -2 \\ 1 & 2 & 1 & -2 \end{bmatrix}$. 其阈值均为 0，该 Hopfield 网，可根据 b 值，选择输出 b_{1i} 或 b_{2i}. 这里称该 Hopfield 网为 **h3**. 可利用 **h3** 实现 select 函数. 图 5 是实现 select 函数的神经网络.

选择函数 A 可由 select 和 equal 函数的复合来定义.

$$A(x_1,x_2,x_3,x_4) = select(equal(x_1,x_2),x_3,x_4)$$

故原始递归函数可用图 6 的神经网络计算得到. 这里 S 为后继函数.

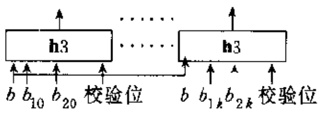

图5 select函数神经网络

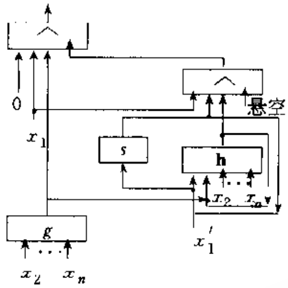

图6 原始递归函数神经网络

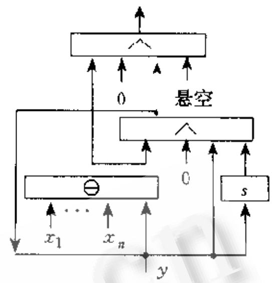

图7 无界搜索神经网络

计算原始递归函数时，先判断 x_1，当 $x_1=0$ 时，g 的值即为所求函数值. 当 $x_1>0$ 时，则进行迭代.

迭代开始时，x'_1 设置为 0，且 g 的值也作为一个初值，送给 **h**. 由于神经元计算需要时间，可在反馈值到达之前将初值撤去，在迭代计算中，当 x'_1 未增加到 x_1 指定值时，应输出函数 Λ 的第 4 个变元，由于函数 Λ 的第 4 个变元无输入，所以系统无输出(即无定义)，而当 x'_1 增加到指定值时，即输出该函数值.

(6)无界搜索(μ-算子)

$$\psi(x_1,\ldots,x_n)\triangleq\mu y[\theta(x_1,\ldots,x_n,y)\downarrow=0\ \&\ (\forall z\leqslant y\ \theta(x_1,\ldots,x_n,z)\downarrow)]$$

无界搜索(μ-算子)可由图 7 的神经网络完成.

计算开始时 y 被置为 0，即 $y=0$，y 在反馈到达之前被撤消. S 为后继函数. 当 θ 和 S 的结果均计算完毕才计算 Λ.

因部分递归函数与图灵机等价，故由定理 1 及定理 2 可得，Hopfield 网与图灵机等价. □

2 结束语

神经网络具有什么样的计算能力，即说它能够有效地计算哪些问题. 本文主要从可计算性角度来讨论 Hopfield 网的能力，证明了 Hopfield 网在计算能力上与图灵机等价.

参考文献

1 Minsky M L. Computation: finite and infinite machines. Prentice-Hall, Inc., 1967

2 Sun *et al*. Turing equivalence of neural networks with second order connection weights. In: Proceedings of the International Joint Conference on Neural Network. Feb. 1991. 357~362

3 Liu Xiao-hong, Dai Ru-wei. Turing equivalence of neural networks of linear-threshold-logic units. Chinese Journal of Computers, 1995, 18(6): 438~442

4 Hopfield J J. Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Science. USA 79, 1982. 2554~2558

5 Liu Xiao-hong, Dai Ru-wei. General methods of construction of Hopfield neural networks. Acta Automatica Sinica, 1996, 22(3): 301~307

6 Cutland N. Computability: an introduction to recursive function theory. London: Cambridge University Press, 1980

The Equivalence of the Hopfield Neural Networks and Turing Machine

MENG Xiang-wu[1,2] CHENG Hu[1]

[1](*Institute of Software The Chinese Academy of Sciences Beijing* 100080)

[2](*Department of Computer Engineering Beijing University of Posts and Telecommunications Beijing* 100876)

Abstract In this paper, the partial recursive function is constructed by Hopfield neural networks. The partial recursive function is equivalent with Turing machine, the computability of Hopfield neural networks is therefore equivalent with Turing machine.

Key words Hopfield neural networks, Turing machine, computability, partial recursive function, neural networks.

Class number TP18