

分布式虚拟环境中基于神经网络的实时预测*

寿黎但 史烈 石教英

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

(浙江大学计算机科学与工程学系 杭州 310027)

摘要 在分布式虚拟环境中,性能的瓶颈是为维护主机间实体行为的一致性而进行的通信.该文针对虚拟场景中的难预测对象建立其状态向量的神经网络模型,使用了基于函数型连接的神经网络对其行为进行实时预测.文章首先介绍了函数型连接的原理和特点;其次,在对传统的 DR (dead reckoning)算法进行描述之后提出了基于神经网络预测的自适应 DR 算法;然后给出了基于该算法的网络软件结构;最后,对一个特例进行了测试,实验结果表明,该算法可以很好地工作.

关键词 分布式虚拟环境,计算机网络,Dead Reckoning (DR 算法),神经网络,函数型连接.

中图法分类号 TP391

分布式虚拟环境(distributed virtual environment)是指,在一组以网络互联的计算机上同时运行虚拟环境(VE)系统的技术.它是虚拟环境与网络技术学科交叉的产物,可以使人们自由地在虚拟环境中实现交互而不受其实际地理位置的制约,从而可在许多科研和应用领域中使用,如远程学习、计算机支持下的协同工作(CSCW)、分布式模拟(distributed simulation)等各种应用.分布式虚拟环境的理论和概念还派生出了许多新的领域,如远程机器人(telerobotics)、远程诊治(telemedicine)等.分布式虚拟环境是目前最热门的研究领域之一,在科学研究、模拟训练、战场仿真等许多方面都有广阔的应用前景.

在分布式虚拟环境中,最根本的要求是每时每刻都要保持分布式系统各主机间场景和场景中各虚拟实体行为的一致性,以维护整个系统的统一.因此需要各主机之间实时地交换大量的状态信息,由此而引起的通信量往往占分布式虚拟环境实体间总通信量的大部分.我们称这一类通信为一致性通信.称一致性通信数据的发送者为参考系统(或标准系统),将接收者称为从属系统.在场景复杂度较高(如数万~数十万实体)或系统规模较大(数百~数千台主机)的情况下^[1],一致性通信往往会导致网络环境的拥塞乃至崩溃.因此,降低一致性通信量是分布式虚拟环境中的一个重要研究课题.

近年来,已有许多研究工作在该方向上展开^[2,3],其中常用的方法是采用自动控制理论中的 DR 算法,对虚拟场景中的实体行为进行预测,减少一致性通信的次数,从而达到降低网络带宽的目的.它们的基本思想是,在参考系统和从属系统上采用完全一致的预测算法,根据虚拟实体的状态历史,对下一时刻的状态进行预测.如果预测结果与实际状态相差不远,则在参考系统端不发送下一时刻的一致性数据,而从属系统中下一时刻的状态就用预测所得的结果来代替;反之,若与实际相差较远,则发送实际状态的数据,即进行一致性通信.这类预测所工作的对象一般有一个共同的特征,即其状态向量可用一个简单的时间函数来表示.但是,实际的虚拟场景中往往存在许多实体,其状态向量无法用一个简单的函数来表示.这类对象一般较智能化,直接受到人的控制,因此,其随意性很大.对于这类实体,传统的 DR 算法已无能为力,这就需要另辟途径来对其进行研究.在本文中,我们使用基于函数型连接的神经网络来对虚拟实体进行实时的预测,得到了较好的效果,大大降低了一致性通信量

* 本文研究得到国家自然科学基金资助.作者寿黎但,1974年生,硕士,主要研究领域为CSCW,多媒体,大规模分布式虚拟环境.史烈,1964年生,讲师,主要研究领域为多媒体,虚拟现实.石教英,1937年生,教授,博士生导师,主要研究领域为计算机图形学,计算机体系结构,虚拟现实.

本文通讯联系人:寿黎但,杭州 310027,浙江大学 CAD&CG 国家重点实验室

本文 1998-06-02 收到原稿,1998-09-16 收到修改稿

(为了避免混淆,我们用网络一词来表示多台主机之间的计算机网络,而用神经网络一词来表示抽象的人工神经网络).

1 函数型连接简介

在函数型连接模型中,一旦一个节点,例如,节点 k 被激励,就会有許多附加的函数功能也被激励,即不仅能得到节点 k 的输出 o_k ,而且还能得到 $f_0(o_k), f_1(o_k), \dots, f_n(o_k)$ ^[4].对函数型连接的具体要求不同,产生的效果也会不同.在本文中,我们考虑的是函数的展开模型和张量(或称外积)模型.

在函数展开模型中,函数型连接单独作用在每个节点上.对于输入模式中的每个节点,它产生相同的附加函数.在这种情况下,输入 x 可以简单地展开成它的幂函数,或者是展开成 n 维空间的正交基函数的一个子集,例如, $\sin \pi x, \cos \pi x, \sin 2\pi x, \cos 2\pi x$ 等等.该神经网络的作用是将输入模式映射到一个更大的模式空间.我们将输入分量与其展开的量联系起来,并用它们来表达输入.这样做,虽然没有引入本质性的新信息,但模式表达却可以得到增强.

在张量或外积模型中,输入模式的每个分量都被乘上整个输入模式矢量.在这种情况下,原来由分量集合 $\{x_i\}$ 所描述的模式变成 $\{x_i, x_i x_j\}$ (其中 $j \geq i$) 或 $\{x_i, x_i x_j, x_i x_j x_k\}$ (其中 $j \geq i, k \geq j \geq i$) 等等.这样做虽然没有引入新的信息,但是它可使神经网络得到联合激励.

在这两种模型中,使表达增强的运算可以适当地混用.通过函数变换后的神经网络由于其输入而得到了增强,仅使用 δ 规则和单层神经网络就可使学习过程迅速收敛^[5].对输入节点个数不超过 5,样本个数较少(10 个以内)的小规模学习问题,只需不到 100 次即可使系统误差小于 10^{-6} .例如,考虑一个三输入(x_1, x_2, x_3),单输出(y)的学习问题,7 组学习样本见表 1.

可以采用混合型函数展开项 $x_1, x_2, x_3, x_1 \cdot x_2, x_1 \cdot x_3, x_2 \cdot x_3, \cos(3\pi \cdot x_2), \cos(3\pi \cdot x_3), \cos(18\pi \cdot x_3 \cdot x_2), \cos(13\pi \cdot x_1 \cdot x_3), \cos(13\pi \cdot x_1 \cdot x_2), \cos(24\pi \cdot x_1 \cdot x_3), \cos(24\pi \cdot x_2 \cdot x_3), \cos(24\pi \cdot x_1 \cdot x_3)$ 等 14 项,采用单层拓扑结构即可使神经网络迅速收敛,如图 1 所示.

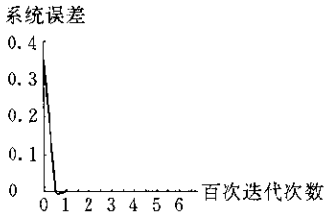


图1 三因素函数型网络的学习速度

表 1 三输入单输出学习样本集

样本序号	x_1	x_2	x_3	y
1	0.216	0.1	0.1	0.90
2	0.347	0.16	0.1	0.85
3	0.463	0.24	0.1	0.80
4	0.487	0.32	0.3	0.75
5	0.513	0.40	0.3	0.7
6	0.547	0.46	0.6	0.65
7	0.780	0.9	0.6	0.6

从图 1 可以看出,在多因素情况下采用扩展的输入项,可使学习的收敛速度大大提高,同时,该神经网络采用了单层的结构,设计简单,编程方便,学习时间极短(在 Pentium 200MHz 机器上不到 1ms),因此,可以满足实时预测的需求.

2 基于函数型连接的预测算法

在分布式虚拟环境中,已经有许多有关虚拟实体状态向量的预测算法的研究.这类算法统称为 DR 算法.一个典型的 DR 预测算法是在参考系统和从属系统两端都使用 Kalman 预测器^[6],根据其运动的状态方程在每一次时钟周期时检查其预测结果与实际状态的误差,如果该误差超过一个约定的阈值,则进行一致性通信,否则省去该次时钟周期内的一致性通信.

在动态的虚拟场景中,许多运动实体常常遵循一定的规律运动.但是也有一些实体的运动没有明显的规律可循,即无法用状态方程表示其运动.举一些简单的例子,如鼠标的移动、数据手套的动作等.这些对象的运动往往受到智能型用户的直接控制,因而其状态改变的随意性较大.换句话说,这类对象的状态较难预测.如果将这

类对象中的某一个看做一个系统,则往往不可能找到一个确定的状态方程来描述它.此时,我们可以求助于神经网络.

在第 1 节中,我们已简单地介绍了基于函数型连接的神经网络.这种神经网络的优越性使得它可胜任实时的预测.算法的基本原理是:对于难预测对象而言,在长时间内,其运动状态往往显得较为混乱而无规律,但是我们有理由认为(算法的实验也确实表明了这一点),在一段短时间内,对象的运动遵循某种难以确定的规律,这种规律是随着时间而改变的,但是状态对于时间的变化总是连续函数,因此,在有限的自变量范围内,根据有限个样本总可以用若干个确定的正交基函数的线性表示来逼近^[5].这也是函数型连接的理论基础.这些正交基函数正是函数型连接中的函数展开项.例如,提供 k 组样本集合 $\{(X_m, X_{m+1}, \dots, X_{m+r-1}, y_{m+r}), (X_{m+1}, X_{m+2}, \dots, X_{m+r}, y_{m+r+1}), (X_{m+k-1}, X_{m+k}, \dots, X_{m+k+r-2}, y_{m+k+r-1})\}$, 其中 y_{m+i} 恰好就是 X_{m+i} ($i=r, \dots, k+r-1$), 则可以选择适当的神经网络,经过学习后使得神经网络输出向量 y'_{m+i} 与样本集合中的 y_{m+i} 一致.因此可以看出,这里的每组样本实际上是根据前 r 个状态,预测第 $r+1$ 个状态.由该组样本训练而成的神经网络可以用来预测 y_{m+k+r} 的值.

由于对象状态不停地改变,这就需要实时地对预测器进行调整,包括修改学习样本集并进行再学习.这样,我们可以在传统的 DR 算法的基础上得到另一种算法,这种算法可以称为基于学习的自适应 DR 算法.由于自适应 DR 算法能对运动方程不确定的对象进行预测,因此特别适用于难预测对象.

在给出算法描述之前我们先对符号表示作一些约定:我们用 k 维向量 $X = (x_1, x_2, \dots, x_k)^T$ 表示被考察对象的状态向量;用 X_t 表示 t_n 时刻对象的状态向量.第 i 组学习样本用含有 $r+1$ 个状态向量的矩阵 $S^{(i)} = (X_{m_i}, X_{m_i+1}, \dots, X_{m_i+r-1}, Y^{(i)})$ 表示,其含义是前 r 个状态向量预测产生函数型连接的输出状态向量 $y^{(i)}$.学习样本集 η 如下: $\{S^{(i)} | i=1, 2, \dots, N\}$.

整个系统分为参考端(标准端)和从属端.参考端的预测算法如下.

算法 1. 参考端的预测算法

Step1. 用伪随机数构造样本集 $\eta = \{S^{(i)} | i=1, 2, \dots, N\}$.

Step2. 用该组样本集对神经网络进行训练学习.

Step3. 将过去的 r 个时间点的状态历史序列 $X_{n-r+1}, X_{n-r+2}, \dots, X_n$ 作为函数型神经网络的 r 个输入,计算输出向量(即预测结果) \tilde{X}_{n+1} .

Step4. 接收到下一个时间同步信号(t_{n+1} 时刻),如果 $\|X_{n+1} - \tilde{X}_{n+1}\| > \epsilon$,则将新状态向量 X_{n+1} 的数据通过计算机网络环境发送到预测端,并将状态 X_{n+1} 加入到历史序列,同时用新样本

$$S = (X_{n-r+1}, X_{n-r+2}, \dots, X_n, X_{n+1})$$

代替 η 中最老的样本;否则,不发送数据,而将预测向量 \tilde{X}_{n+1} 当做 t_{n-1} 时刻的状态加入到状态历史序列之中,同时用

$$S = (X_{n-r+1}, X_{n-r+2}, \dots, X_n, \tilde{X}_{n+1})$$

更新样本集 η .

Step5. 无条件跳转至 Step2,依次无限循环执行 Step2~Step4 各步操作.

由算法的描述可知,每收到一个时钟信号,算法就循环一次.在上述算法的 Step3 中需要忽略最初的 r 次循环,因为在最初的 r 次循环时不存在长度为 r 的历史序列.另外,每次循环都要改变相应序列的下标.这些说明同样也适用于从属端的算法.从属端的算法可以表示如下.

算法 2. 从属端的预测算法

Step1. 用与算法 1 相同的伪随机数构造样本集 $\eta = \{S^{(i)} | i=1, 2, \dots, N\}$.

Step2. 用该组样本集对神经网络进行训练学习.

Step3. 将过去的 r 个时间点的状态历史序列 $X_{n-r+1}, X_{n-r+2}, \dots, X_n$ 作为函数型神经网络的 r 个输入,计算输出向量(即预测结果) \tilde{X}_{n+1} .

Step4. 接收到下一个时间同步信号(t_{n+1} 时刻),如果在此之前收到参考端的数据包,则将收到的新状态

X_{n+1} 加入历史序列,同时用新样本

$$S = (X_{n-r+1}, X_{n-r+2}, \dots, X_n, X_{n+1})$$

代替 η 中最老的样本;否则,即未收到数据包,则将预测向量 \tilde{X}_{n+1} 当做 t_{n+1} 时刻的状态加入状态历史序列,同时用

$$S = (X_{n-r+1}, X_{n-r+2}, \dots, X_n, \tilde{X}_{n+1})$$

更新样本集 η .

Step5. 无条件跳转至 Step2,依次无限循环执行 Step2~Step4 各步操作.

3 系统的软件结构

根据上节中所给出的算法,我们可以设计出相应的网络软件结构,如图2所示.

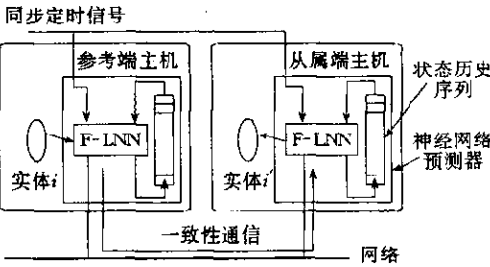


图2 基于神经网络预测的系统软件结构

对应于一个实体 i ,参考端主机上有一个神经网络预测器,其内部运行算法1;与它相对应,从属端主机上必然存在一个相应的从属实体 i' 和神经网络预测器,该预测器执行算法2(图中 F-LNN 是基于函数型连接的神经网络的缩写).由于分布式虚拟环境中主要的性能限制来自网络带宽瓶颈,从该软件结构图可以看出,采用神经网络预测器的实质是用消耗计算资源的方法来降低对网络资源的需求.基于函数型连接的神经网络具有极高的收敛速度(多数情况 $< 1ms$),因此当实体个数

较大时,预测过程所消耗的单机计算资源仍然不多.可见,该预测算法提供了良好的可扩展性,对于目前计算机领域内很热门的大规模分布式虚拟环境^[3](large-scale distributed virtual environment)具有参考价值.

4 实验结果

由于我们所关心的实体行为是“难以预测”的,实体行为的宏观表现较为混乱,所以往往不需要保留很长的状态历史序列.一般只需不到10个.另外,我们所考察的虚拟实体在虚拟空间中函数型连接的学习过程非常迅速,在样本个数不超过10个时,只需100~200次循环学习即可使神经网络收敛.当输入的状态向量个数不超过5个时,这样的计算量在目前的高档PC机上只需不到1ms的时间,因此该算法完全可以用于实时的预测.

我们对空间自由度为2的鼠标进行了实验,硬件平台是用10Mbps的以太网互连的20台Pentium 200MHz的PC机,软件平台采用MS Windows95,实验对象是4名熟练用户.另外,我们对同样的数据使用二次多项式插值的预测作为对照.在该实验的神经网络预测算法中, r 值取为2,样本集所含的样本个数 N 为4.因此,我们可以建立含有两个输入向量,一个输出向量的函数型神经网络.设两个输入向量分别为 (x_{n-1}, y_{n-1}) 和 (x_n, y_n) ,则对这两个向量作如下19维的混合型函数展开: $x_{n-1}, y_{n-1}, x_n, y_n, x_{n-1} \cdot y_{n-1}, x_{n-1} \cdot x_n, y_{n-1} \cdot x_n, x_n \cdot y_n, x_n \cdot y_{n-1}, x_{n-1} \cdot y_n, x_n \cdot y_n, \cos(3\pi \cdot x_{n-1}), \cos(3\pi \cdot y_{n-1}), \cos(3\pi \cdot x_n), \cos(3\pi \cdot y_n), \cos(8\pi \cdot x_{n-1} \cdot y_{n-1}), \cos(8\pi \cdot x_{n-1} \cdot y_n), \cos(13\pi \cdot x_n \cdot y_n), \cos(13\pi \cdot x_n \cdot y_{n-1})$,并实时预测新的状态向量 (x_{n+1}, y_{n+1}) ,结果如图3所示.

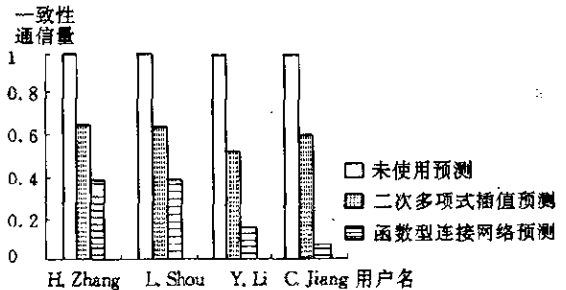


图3 预测算法对4个用户的鼠标使用的结果

实验结果表明,该算法对所有用户所操纵的鼠标都能很好地进行预测,而且该算法的预测精度优于多项式插值的预测算法.产生这种结果的原因是由于神经网络预测器能够实时地修改其权值,使之在不同的时刻进行

自适应. 在模拟实体数目较大的情况下(例如, 10^3 个), 算法可以节省 50%~70% 的一致性通信量, 对所有实体进行一次预测消耗 CPU 时间 2.4s~3.0s. 但是对于大规模系统(实体数 $>10^5$) 来说, 模拟测试表明, 消耗在预测算法上的计算量非常巨大, 这就需要强大的单机计算能力的支持. 另外, 对于传统 DR 算法的预测对象, 即状态可表示为显式时间函数的实体, 该神经网络算法也可很好地进行预测, 本文不再赘述.

5 结 论

(1) 函数型连接是一种采用函数并行激励机制的神经网络, 这种神经网络的特点是采用单层拓扑结构和 δ 学习规则, 因而收敛速度非常高. 本文给出了一个基于函数型连接的神经网络预测算法用于对分布式虚拟环境中的实体状态进行实时的预测, 克服了传统的 DR 算法对难预测对象无法建立其控制论模型的缺点. 实验结果表明, 对自由度为 2 的鼠标, 该算法可以很好地工作.

(2) 在分布式虚拟环境中, 主要的性能瓶颈是网络带宽. 文中所述的基于函数型连接的神经网络预测算法降低了网络带宽的消耗, 有助于提高整个系统的性能和可扩展性.

参 考 文 献

- 1 Pentland A P. Computational complexity versus simulated environments. *Computer Graphics*, 1990, 24(2): 185~192
- 2 Pratt D R. A software architecture for the construction and management of real time virtual environments [Ph. D. Thesis]. Monterey, California: Naval Postgraduate School, 1993
- 3 Macedonia M R. A network software architecture for large scale virtual environments [Ph. D. Thesis]. Monterey, California: Naval Postgraduate School, 1995
- 4 包约翰(美). 自适应模式识别与神经网络. 北京: 科学出版社, 1992
(Pao Yoh-han. Adaptive Pattern Recognition and Neural-networks. Menlo Park, MA: Addison-Wesley Publishing Company, Inc., 1989. 192~215)
- 5 Pao Yoh-han, Yoshiyasu Takefuji. Functional-link net computing: theory, system architecture, and functionalities. *Computer*, 1992, 25(5): 76~79
- 6 Baumgartner, Eric T, Skaar, Steven B. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 1994, 39(3): 493~502

Real-time Prediction Based on Neural-networks in Distributed Virtual Environment

SHOU Li-dan SHI Lie SHI Jiao-ying

(State Key Laboratory of CAD & CG Zhejiang University Hangzhou 310027)

(Department of Computer Science and Engineering Zhejiang University Hangzhou 310027)

Abstract In a distributed virtual environment (DVE), the bottleneck of performance is the communication among hosts which keeps the consistency of virtual entities. The neural-network models of state vectors for unpredictable entities in a virtual scene are built in this paper, and the functional-link net to real-time prediction of their behavior is applied. Firstly, the principles and characters of the functional-link net are introduced in this paper. Secondly, after description of the traditional dead reckoning (DR) algorithm, an adaptive version of the algorithm based on the functional-link net is presented; the network software architecture based on the algorithm is also given. Finally, an example of the algorithm is given with experimental data, which shows the good performance of it.

Key words Distributed virtual environment, computer networks, dead reckoning, neural-network, functional-link net.