利用反向传播算法合理分配缓冲区

吴援明[①]　高　科[①]　李乐民[②]
[①](电子科技大学光电信息学院　成都　610054)
[②](电子科技大学宽带光纤传输与通信系统技术重点实验室　成都　610054)

摘　要　该文提出了一种新的缓冲区分配方法，即动态神经共享 (Dynamic Neural Sharing，DNS)方法。这种方法利用反向传播算法合理分配缓冲区资源，从而减少自相似业务的分组丢失率。通过两组仿真实验发现，与完全分割(Complete Partitioning, CP), 完全共享(Complete Sharing, CS), 部分共享(Partial Sharing, PS)这些传统的缓冲区分配方法相比，DNS 在减少分组丢失和体现公平性(每个源都占有一定数量的缓冲区资源)之间达到了更好的平衡。

关键词　反向传播算法，分组丢失，DNS，自相似，ON-OFF 源
中图分类号：TN915.07　　**文献标识码**：A　　**文章编号**：1009-5896(2006)08-1418-04

Managing Buffer Appropriately with the Back Propagation Learning Algorithm

Wu Yuan-ming[①]　Gao Ke[①]　Li Le-min[②]
[①](*School of Optoelectronic Information, UEST of China, Chengdu* 610054, *China*)
[②](*Key Laboratory on Broad band Fiber-Optical Transmission & Communication Network Technology, UEST of China, Chengdu* 610054，*China*)

Abstract　A novel buffer management algorithm named DNS(Dynamic Neural Sharing) is suggested in this paper. This algorithm utilizes the Back Propagation learning Algorithm(BPA) to manage buffer appropriately, thus reduce the packet loss in self-similar teletraffic patterns. A conclusion is drawn through two emulations that the DNS addresses the trade off between packet loss and fairness issues better than those traditional algorithms such as CP(Complete Partitioning), CS(Complete Sharing) and SPS(State Partial Sharing).

Key words　Back Propagation learning Algorithm(BPA), Packet loss, Dynamic Neural Sharing (DNS), Self-similar, ON-OFF source

1　引言

神经网络是一种非线性系统，它能够像人一样，通过学习(亦即训练)来达到一定的功能，从而完成相应的任务。神经网络的一个显著特点是：如果系统中存在可以接受的误差，那么这个误差不会影响整个系统的正确运行，依然能够得到好的结果[1]。这一特点使得神经网络模型比其他模型更适合于具有突发特性的自相似业务流，再加上简单的学习方法，使得它可以作为系统业务流模拟的一种强有力的工具。所以，有一些学者利用神经网络来解决通信网系统中的一些非线性问题：Yuang等人提出了利用反向传播神经网络来解决无线ATM网络带宽分配问题[2]；Bolla等人提出了利用神经网络的预测能力来解决包含同步电路交换和异步分组交换时分复用混合结构中的带宽分配问题[3]；Long提出了利用神经网络的预测能力来控制状态变化未知的系统结构[4]；Mohamed提出利用神经网络的预测能力来自动量化视频数据流的质量[5]；Yousefi'zadeh等提出利用神经网络的预测功能来减少具有自相似特性的业务模型的分组丢失[1]。

本文采用 5 层神经网络结构(见图 1)和反向传播算法，利用神经网络的学习能力和预测能力，为自相似业务流建

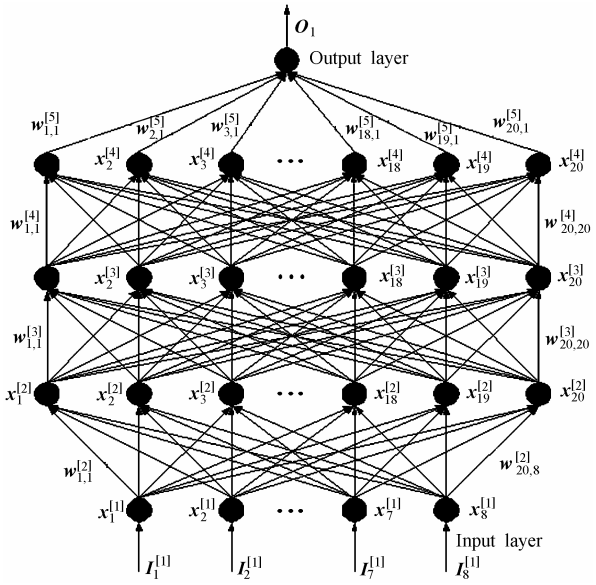

图 1　神经网络结构

Fig.1　The structure of neural network used for the task of modeling

2004-12-06 收到，2005-06-27 改回
国家 863 计划项目(2002AA122021)和电子科技大学青年科技基金(YF020504)资助课题

模，来预测业务流的到达过程，比较 4 种缓冲区分配方案，根据分组丢失率的大小，对 ON-OFF 模型产生的自相似业务流及来源于贝尔实验室的实测网络数据流进行仿真，得到 DNS 缓冲区分配方案的丢失率和公平性优于 CP，CS，PS 缓冲区分配方案。为了提高预测能力，减小预测误差，对这些数据进行相应的平滑处理。为了提高反向传播算法的收敛速度，采用了动量因子平滑收敛过程中的振荡，使得算法在稳定的前提下更快收敛。

2　利用改进的反向传播算法实现对自相似业务流的预测

本文仿真所用神经网络共 5 层结构，1 个输入层(8 个神经元)，3 个隐层(每个隐层中有 20 个神经元)，1 个输出层(1 个神经元)。每 1 层的所有神经元都要经过加权，再总体相加，然后经过传输函数作为下一层神经元的输入。

用 $\boldsymbol{x}_j^{[s]}$ 表示 s 层第 j 个神经元的当前输出状态，$\boldsymbol{w}_{ji}^{[s]}$ 表示第 $(s-1)$ 层的第 i 个神经元与第 s 层的第 j 个神经元之间的连接权值，$\boldsymbol{I}_j^{[s]}$ 表示第 s 层的第 j 个神经元的净输入，传输函数 $f(z)=1/\left(1+e^{-z}\right)$，$\boldsymbol{e}_j^{[s]}$ 表示第 s 层第 j 个神经元与第 $(s+1)$ 层之间的反向传播算子，$\boldsymbol{d}_k$ 表示输出层第 k 个神经元的期望输出，$\boldsymbol{O}_k$ 表示输出层第 k 个神经元的实际输出，$\boldsymbol{e}_k$ 表示输出层第 k 个神经元的反向传递算子，lc 表示学习速度，为了提高收敛速度还加入了动量因子 M。根据

$$\boldsymbol{I}_j^{[s]}=\sum_i\left(\boldsymbol{w}_{ji}^{[s]}\boldsymbol{x}_i^{[s-1]}\right) \tag{1}$$

$$\boldsymbol{x}_j^{[s]}=f\left(\boldsymbol{I}_j^{[s]}\right)=f\left\{\sum_i\left(\boldsymbol{w}_{ji}^{[s]}\boldsymbol{x}_i^{[s-1]}\right)\right\} \tag{2}$$

由反向传递算子的定义可知

$$\boldsymbol{e}_j^{[s]}=-\frac{\partial E}{\partial I_j^{[s]}} \tag{3}$$

式中 E 为绝对误差函数。

利用偏倒数的链式法则，将式(1)和式(2)代入式(3)，可得

$$\begin{aligned}\boldsymbol{e}_j^{[s]}&=\sum_k\frac{\partial E}{\partial \boldsymbol{I}_k^{[s+1]}}\frac{\partial \boldsymbol{I}_k^{[s+1]}}{\partial \boldsymbol{I}_j^{[s]}}=\sum_k\boldsymbol{e}_k^{[s+1]}*\frac{\partial\sum_i\boldsymbol{w}_{ki}^{[s+1]}\boldsymbol{x}_k^{[s]}}{\partial \boldsymbol{I}_j^{[s]}}\\&=\sum_k\boldsymbol{e}_k^{[s+1]}\frac{\partial\sum_i\boldsymbol{w}_{ki}^{[s+1]}f\left\{\boldsymbol{I}_k^{[s]}\right\}}{\partial I_j^{[s]}}=\sum_k\boldsymbol{e}_k^{[s+1]}\boldsymbol{w}_{kj}^{[s+1]}f'\left\{\boldsymbol{I}_j^{[s]}\right\}\\&=f'\left\{\boldsymbol{I}_j^{[s]}\right\}\sum_k\left\{\boldsymbol{e}_k^{[s+1]}\boldsymbol{w}_{kj}^{[s+1]}\right\}\end{aligned} \tag{4}$$

其中 f' 表示 f 的导数，且 f 是处处连续函数，我们取

$$f(z)=\frac{1}{1+e^{-z}} \tag{5}$$

$$f'(z)=\frac{e^{-z}}{(1+e^{-z})^2}=\frac{1}{(1+e^{-z})}\left(1-\frac{1}{1+e^{-z}}\right)=f(z)\{1-f(z)\}$$

所以

$$f'\left(\boldsymbol{I}_j^{[s]}\right)=f\left(\boldsymbol{I}_j^{[s]}\right)\left\{1-f\left(\boldsymbol{I}_j^{[s]}\right)\right\}=\boldsymbol{x}_j^{[s]}\left\{1-\boldsymbol{x}_j^{[s]}\right\} \tag{6}$$

将式(6)带入式(4)得

$$\boldsymbol{e}_j^{[s]}=\boldsymbol{x}_j^{[s]}\left\{1-\boldsymbol{x}_j^{[s]}\right\}\sum_k\left\{\boldsymbol{e}_k^{[s+1]}\boldsymbol{w}_{kj}^{[s+1]}\right\} \tag{7}$$

$$\boldsymbol{e}_k=\left(\boldsymbol{d}_k-\boldsymbol{O}_k\right)\boldsymbol{O}_k\left(1-\boldsymbol{O}_k\right) \tag{8}$$

$$\underbrace{\Delta\boldsymbol{w}_{ji}^{[s]}}_{\text{第}(k+1)\text{步}}=\text{lc}\boldsymbol{e}_j^{[s]}\boldsymbol{x}_i^{[s-1]}+M\underbrace{\left(\Delta\boldsymbol{w}_{ji}^{[s]}\right)}_{\text{第}k\text{步}} \tag{9}$$

本文使用批处理算法，将一批训练样本进行训练后才进行一次网络权值的更新，也即将每个神经元梯度 $\left(\boldsymbol{e}_j^{[s]}\boldsymbol{x}_i^{[s-1]}\right)$ 的平均值 $\left(\text{mean}\left(\boldsymbol{e}_j^{[s]}\boldsymbol{x}_i^{[s-1]}\right)\right)$ 作为式(9)中的梯度进行网络权值的更新。那么，式(9)变为

$$\underbrace{\Delta\boldsymbol{w}_{ji}^{[s]}}_{\text{第}(k+1)\text{步}}==\text{lcmean}\left(\boldsymbol{e}_j^{[s]}\boldsymbol{x}_i^{[s-1]}\right)+M\underbrace{\left(\Delta\boldsymbol{w}_{ji}^{[s]}\right)}_{\text{第}k\text{步}} \tag{10}$$

经过改进的反向传播算法的步骤可以描述为：

(1)将一批训练样本从输入层前向传播到输出层，在这个传播的过程中，每次训练所有神经元的净输入和输出状态都可以确定。

(2)对于每次训练来说，输出层的每个神经元计算局部误差，并根据式(8)计算输出层的反向传播算子，将这批训练输出层每个神经元的梯度平均在一起，然后根据式(10)更新输出层权值。

(3)每次训练中，每一层的神经元都根据式(7)在后一层的基础上计算反向传播算子，并求出每个神经元的平均梯度，根据式(10)更新权值。

反向传播算法的时间复杂度为 $O(IN)$，空间复杂度为 $O(N)$，其中I代表迭代次数，N代表网络中权值的个数[1]。

3　4 种缓冲区分配方案

4 种缓冲区分配方案分别是完全分割(CP)方案、完全共享(CS)方案、部分共享(PS)和动态神经共享(DNS)方案。在本文中，CP 方案根据系统中数据业务流的个数将缓冲区均分，固定分配给每个业务流，缓冲区没有共享；CS 方案将缓冲区作为一个可以共享的整体被各个业务流使用，每个业务流没有固定的缓冲区大小；SPS(Static Partial Sharing)方案将缓冲区分成两个部分，整个缓冲区的 2/5 作为共享缓冲区，剩下的 3/5 作为固定缓冲区被平分给每个业务流；DNS 方案也是将缓冲区分成两个部分，整个缓冲区的 2/5 作为共享缓冲区，而剩下的 3/5 则按照神经网络的预测结果在各个业务流之间进行动态分配。本文采用 2 个业务流进行仿真，并设定不同的优先级，且认为业务流与对应缓冲区之间是先来先服务的系统。

4　基于预测结果降低分组丢失率的原理及算法

(1)本仿真选用 2 个源($\boldsymbol{X}$，$\boldsymbol{Y}$)，由于神经网络传输函数为 $f(z)=1/(1+e^{-z})$，其值域为(0，1)，因此训练集中数据分布在(0，1)内。DNS 方法过程描述如下：如果用 a 表示每次训

练的样本数(训练所需样本数取决于业务模型的复杂度)，每次预测 20 个数据，以(a+20)为区间分别对 $\boldsymbol{X}$，$\boldsymbol{Y}$ 进行平滑，得到两组训练集 $\boldsymbol{X}'$，$\boldsymbol{Y}'$。$\boldsymbol{X}'$，$\boldsymbol{Y}'$ 的计算公式如下：

$$\boldsymbol{X}'(i)=\frac{\sum_{k=1}^{i}\boldsymbol{X}(k)}{(a+20)\times\max(\boldsymbol{X})},\ i=1,2,\cdots,(a+20);\ \ \max(\boldsymbol{X})$$

表示 $\boldsymbol{X}$ 数组的最大值。

$$\boldsymbol{Y}'(i)=\frac{\sum_{k=1}^{i}\boldsymbol{Y}(k)}{(a+20)\times\max(\boldsymbol{Y})},\ i=1,2,\cdots,(a+20);\ \ \max(\boldsymbol{Y})$$

表示 $\boldsymbol{Y}$ 数组的最大值。

(2)分别对 $\boldsymbol{X}'$，$\boldsymbol{Y}'$ 进行训练。对 $\boldsymbol{X}'$ 进行第t次训练过程的训练集{$\boldsymbol{P}_t$，$\boldsymbol{d}_t$}亦即{输入向量，目标向量}可表示为

$$\boldsymbol{P}_t^{(j)}=\left[\boldsymbol{X}'(20\times(t-1)+j),\boldsymbol{X}'(20\times(t-1)+j+1),\cdots,\boldsymbol{X}'(20\times(t-1)+j+7)\right]^{\mathrm{T}};$$

$$\boldsymbol{d}_t^{(j)}=\left[\boldsymbol{X}'(20\times(t-1)+j+8)\right]$$

其中 $j=1,2,\cdots,a$；　$t=0,1,\cdots,(b-1)$，T 表示转置。

对 $\boldsymbol{Y}'$ 进行训练也有相似的表达式。每次训练后预测 20 个后继到达分组数，共进行 b 次训练，得到 20×b 个预测数据(a+20×b=17000)构成的预测数据组 $\boldsymbol{X}''$，$\boldsymbol{Y}''$。

(3)分别对 $\boldsymbol{X}''$，$\boldsymbol{Y}''$ 进行与平滑相反的变换得到 $\boldsymbol{X}'''$，$\boldsymbol{Y}'''$，使之分别对应于 $\boldsymbol{X}$，$\boldsymbol{Y}$ 中的第(a+1)~17000 个数据。反平滑公式可表示为

$$\boldsymbol{X}'''(l)=(a+20)\times\max(\boldsymbol{X})\times\boldsymbol{X}''(l)-\sum_{k=1}^{(a+l-1)}\boldsymbol{X}(k)$$

$$l=1,2,\cdots,20\times b$$

$$\boldsymbol{Y}'''(l)=(a+20)\times\max(\boldsymbol{Y})\times\boldsymbol{Y}''(l)-\sum_{k=1}^{(a+l-1)}\boldsymbol{Y}(k)$$

$$l=1,2,\cdots,20\times b$$

有时预测出的结果会小于对应源的最小值或者大于其最大值，可以将小于最小值的数值变为最小值，将大于最大值的数值变为最大值。

5 仿真实验结果及分析

在仿真中，调度算法采用 FTDM(Fixed Time Division Multiplexing)，每个源占用一定的服务器带宽和服务速率，没有带宽的共享。每个源对应的服务器速率分别为每组数据的平均值 mean($\boldsymbol{X}$), mean($\boldsymbol{Y}$)。在有缓冲区共享的方法中，数据产生率低的源在缓冲区共享资源的占用上优先级比较高。不管是缓冲区的管理还是服务器的调度，认为源与对应的缓冲区之间是先来先服务的队列系统。将每个源的到达速率与服务速率相比，若到达速率小于服务速率，就没有分组丢失；若到达速率大于服务速率，缓冲区中的队列就会变长，若队列长度超出了此源可占用缓冲区资源的大小，就会出现分组丢失。

本部分进行两组仿真实验，实验所需参数见表 1。观察当缓冲区的大小从 1 到总服务器速率(3×(mean($\boldsymbol{X}$)+mean($\boldsymbol{Y}$)))变化时(这里缓冲区大小与分组个数一一对应)，CP，CS，SPS，DNS 的分组丢失情况，并进行比较。其中 CP 将缓冲区资源固定均分给 2 个源，SPS 将缓冲区分为两部分，3/5 是共享区域，余下的 2/5 固定均分给 2 个源。通过仿真实验，分别得出总丢失率的比较曲线和优先级低的源的丢失率的比较曲线。

仿真实验 1 中的 $\boldsymbol{X}$ 源数据由 50 个 ON-OFF 源产生自相似业务；$\boldsymbol{Y}$ 源数据由 30 个 ON-OFF 源产生自相似业务，ON 期间和 OFF 期间均服从 Pareto 分布，突发参数 H 值分别为 0.8217 和 0.8094，动量因子分别取为 0.65 和 0.60。仿真实验 2 中的 $\boldsymbol{X}$ 源来源于贝尔实验室 1989 年 10 月 5 日 11：00 开始测量的广域以太网每 0.1s 内到达的分组数，$\boldsymbol{Y}$ 源数据来源于贝尔实验室 1989 年 8 月 29 日 11：25 开始测量的局域以太网(一小部分是广域网传输数据)每 0.1s 内到达的分组数。这些数据都具有自相似性，突发参数 H 值分别为 0.9286 和 0.8145，相应的动量因子分别取为 0.70 和 0.61。

表 1 两组仿真实验所需参数

Tab.1 Parameters for two simulations

	max ($\boldsymbol{X}$)	mean ($\boldsymbol{X}$)	max ($\boldsymbol{Y}$)	mean ($\boldsymbol{Y}$)	a	b	缓冲区最大值
仿真 1	39	25	24	15	2000	750	120
仿真 2	686	59	122	37	2000	750	288

仿真实验结果示于图 2–图 5。

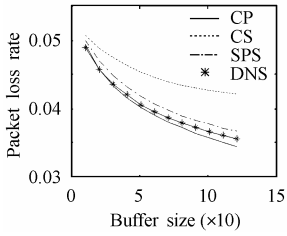

图 2 仿真 1 中优先级低的源的丢失率比较曲线

Fig.2 The comparing curve of lower priority source packet loss in the first simulation

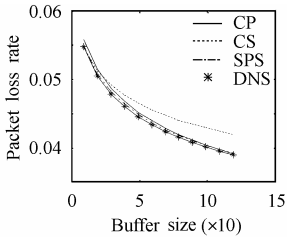

图 3 仿真 1 中总丢失率的比较曲线

Fig.3 The comparing curve of total packet loss in the first simulation

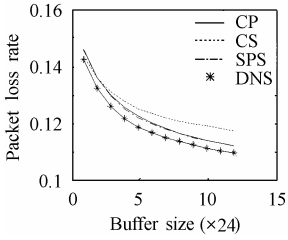

图 4 仿真 2 中优先级低的源的丢失率比较曲线

Fig.4 The comparing curve of lower priority source packet loss in the second simulation

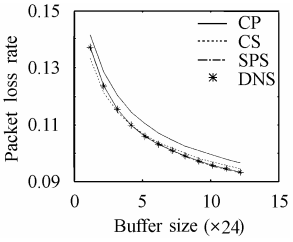

图 5 仿真 2 中总丢失率的比较曲线

Fig.5 The comparing curve of total packet loss in the second simulation

从以上图形可以看出：SPS 方法达到了提高总的分组丢失率和体现公平性之间的平衡，DNS 方法的性能比 SPS 方法的性能又有所提高，得出了满意的实验结果。

同时可以看出，仿真实验 2 的结果与仿真实验 1 相比，分组丢失率比较大。这是因为神经网络训练所需要的样本数取决于业务模型的复杂度，实际的网络复杂度比用 ON-OFF 源模拟的系统复杂度要大得多，需要的训练样本数也要更多。此处进行了比较简单的仿真试验，选取了比较少的训练样本数，其目的只是在 4 种算法之间进行比较，仍然得到了满意的实验结果。要得到数量级小的分组丢失率，只需选用更多的训练样本即可，但这样会增加训练的时间。如果要对更加复杂的网络进行模拟，需要的训练样本数很多，这是非常费时的。这也算是神经网络的一个缺点。

此外，在训练的过程中，有时会出现不收敛的情况。这是由于神经网络对初始值的选取十分敏感，若初始值选取不当，就会导致结果不收敛。在实验中，一般选取初始值为一个比较小的数值(如 0.01)。

6　结束语

本文分别对用 ON-OFF 源模拟的自相似业务模型以及实际的业务模型进行了实验，得出了 DNS 缓冲区分配方案的丢失率和公平性优于 CP，CS，PS 缓冲区分配方案，证明了利用神经网络模型合理分配缓冲区可以减少分组丢失率，并能使得每个源都能合理的占用一定的缓冲区资源，体现了公平性。当然，本文采用的神经网络模型，对初始值的选择、动量因子的选择以及训练算法的收敛速度较慢等，都需要进行更深入的研究。

参 考 文 献

[1] Yousefi'zadeh H, Jonckheere E A, Silvester J A. Utilizing neural networks to reduce packet loss in self-similar teletraffic patterns. IEEE International Conference on Communications, Anchorage, AK, USA. ICC’03, 2003, (3): 1942−1946.

[2] Yuang M C, Tien P L, Chen C S. A contention access protocol with dynamic bandwidth allocation for wireless ATM networks. IEEE International Conference on Communications. New Orleans, USA, ICC2000, 2000, (1): 149−153.

[3] Bolla R, Davoli F, Maryni P,*et al*.. A neural strategy for optimal multiplexing of circuit- and packet-switched traffic. Global Telecommunications Conference, Orlando, Florida, USA, GLOBECOM’92, 1992, (3): 1324−1330.

[4] Long T W. A self-similar neural network for distributed vibration control. Proceedings of the 32nd IEEE Conference on Decision and Control, San Antonio, Texas, USA, 1993, (4): 3243−3248.

[5] Mohamed S, Rubino G. A study of real-time packet video quality using random neural networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 2002, 12, (12): 1071−1083.

吴援明：　男，1966 年生，副教授，研究方向为现代通信中的信号处理技术.

高　科：　女，1981 年生，硕士生，研究方向为现代通信中的信号处理技术.

李乐民：　1932 年生，教授 ，中国工程院院士，研究方向为通信网(包括宽带通信网和移动通信网).