

## 基于双层递归神经网络的路由优化算法

周井泉 张顺颐\*

(南京邮电大学电子工程系 南京 210003)

\*(南京邮电大学校长办公室 南京 210003)

**摘要:** 研究分组交换网的路由选择及流量分配问题,以网络的平均时延为优化目标函数。为使问题的解能实时、可靠地完成,将一种用于最短路径计算的双层递归神经网络应用于路由选择的流量导数法中。仿真结果表明,该算法在收敛的可靠性和计算的实时性方面有所提高。

**关键词:** 神经网络, 优化算法, 流量导数法, 最短路径, 独立变量

中图分类号: TP183

文献标识码: A

文章编号: 1009-5896(2005)12-1901-04

## Routing Algorithm Based on Two-layer Recurrent Neural Network

Zhou Jing-quan Zhang Shun-yi\*

(Department of Electronic Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

\*(Director Office, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract** This paper is concerned with the optimum routing and flow assignment problem in packet-switch networks. The optimum objective function is the network-wide average time delay. To make the solution be implemented reliably in real time, a neural network for shortest path computation that is a two-layer recurrent structure is applied to flow deviation method. Simulation results show that improvements can be achieved in the reliability of successful convergence and in the decrease of computation time.

**Key words** Neural network (NN), Optimization algorithm, Flow deviation (FD) method, Shortest path (SP), Independent variables

### 1 前言

路由和流量分配是影响分组交换网服务质量(QoS)的重要因素。在给定网络拓扑结构和各节点对之间通信量的条件下,合理选择各节点之间的通信路由及分配各链路的流量,可提高网络资源的利用率和服务质量,减小成本。

在以网络中传播数据的总平均时延作为路由优化目标的情况下,路由问题可描述成典型的非线性数学规划问题<sup>[1]</sup>,问题解的循环算法有流量导数法(FD)<sup>[2]</sup>、梯度投影法(GP)<sup>[3]</sup>等等。但这些方法不能有效地应用于需要实时计算的场合。大多数网络层的路由循环算法包含了最短路径(SP)计算,Ali-Kamoun将循环算法中的SP计算用神经网络(NN)来实现,从而解决了实时性<sup>[1]</sup>,该方法需要的神经元数取决于网

络节点的平方。本文的NN中每个神经元对应网络的一条边(或称支路、链路),因而神经元数等于边数,通常少于节点的平方,从而简化NN的设计<sup>[4]</sup>。根据与有向图的各条边相对应的决策变量的独立性,NN由独立变量神经元和非独立变量神经元组成,它们构成双层递归NN。仿真结果表明,该NN与FD方法相结合能提高计算速度和可靠性。

### 2 最短路径的数学描述及独立决策变量

#### 2.1 最短路径的约束关系

对于一个给定的有向图 $G=(V,E)$ ,设其节点数为 $n$ ,边数为 $m$ , $V$ 和 $E$ 分别为这些节点和边的集合。任选一个节点作为参考节点,则余下的节点与支路间的关联可以用一个 $(n-1) \times m$ 阶的关联矩阵表示: $A=[a_{ij}]$ 。其中第 $i$ 行, $j$ 列元

素( $i=1,2,\dots,n, j=1,2,\dots,m$ )

$$a_{ij} = \begin{cases} 1, & \text{如果支路}j\text{与节点}i\text{关联, 且它的方向背离该节点} \\ -1, & \text{如果支路}j\text{与节点}i\text{关联, 且它的方向指向该节点} \\ 0, & \text{如果支路}j\text{与节点}i\text{不关联} \end{cases}$$

对于源点为 $s$ , 终点为 $d$ 的一条路径 $p_{sd}$ , 可用与图中 $m$ 条支路相对应的一组决策变量的取值来表示, 即用 $\mathbf{v}=[v_1, v_2, \dots, v_m]^T$ 表示, 其中各决策变量 $v_i(i=1,2,\dots,m)$ 的取值为0或1, 即: 若第 $i$ 条支路在该路径上时,  $v_i=1$ ; 若第 $i$ 条支路不在该路径上时,  $v_i=0$ .

若将路径 $p_{sd}$ 上的各决策变量的取值1看作是该支路上通过的单位电流, 不在路径上的决策变量的取值0看作是零

$$\phi_i = \begin{cases} 1, & i = s \\ -1, & i = d \\ 0, & \text{其它} \end{cases}$$

电流, 则根据基尔霍夫电流定律, 可得构造一条路径时, 各支路决策变量应满足的约束关系如式(2), 其中 $\Phi$ 是 $n-1$ 阶列向量, 其元素:

因此, 最短路径问题描述如下:

最小化 
$$z = \sum_{i=1}^m c_i v_i = \mathbf{c}\mathbf{v} \tag{1}$$

约束条件 
$$\mathbf{A}\mathbf{v} = \Phi \tag{2}$$

和 
$$v_i \in \{0, 1\}, i=1,2,\dots,m \tag{3}$$

其中 $\mathbf{c}=[c_1, c_2, \dots, c_m]$ 为 $m$ 阶行向量, 各元素为非负的数, 代表了各支路的长度。

### 2.2 独立变量

将计算路径的各决策变量看作是支路电流变量, 由网络理论可知, 连支电流变量是独立的, 独立变量数为 $m-n+1$ 个。而树支电流变量不独立, 可由连支电流变量来表示<sup>[5]</sup>。

选取独立变量时, 各支路的编号作如下处理:

(1)在有向图中任取一棵树, 并将树支的编号取为 $1 \sim n-1$ , 则对应的树支变量 $\mathbf{v}_t=[v_1, v_2, \dots, v_{n-1}]^T$ 是非独立的。

(2)余下的支路为连支, 编号为 $n \sim m$ , 则对应的连支变量 $\mathbf{v}_l=[v_n, v_{n+1}, \dots, v_m]^T$ 为独立变量。

(3)决策变量按先树支后连支的次序编号排列, 分成树支变量和连支变量两部分:  $\mathbf{v} = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_l \end{bmatrix}$ 。相应的关联矩阵分成 $\mathbf{A}=[\mathbf{A}_t, \mathbf{A}_l]$ 。

由式(2)可得

$$\mathbf{v}_t = \mathbf{A}_t^{-1}\Phi - \mathbf{A}_t^{-1}\mathbf{A}_l\mathbf{v}_l \tag{4}$$

## 3 基于决策变量独立性的双层递归神经网络

### 3.1 非独立变量神经网络

利用神经网络求解最短路径问题, 将每条边的决策变量对应一个神经元的状态。与独立变量 $\mathbf{v}_l$ 相对应的神经网络称为独立变量神经网络, 与树支变量 $\mathbf{v}_t$ 相对应的称为非独立变量神经网络。非独立变量神经网络的方程设为

$$v_t = w_t v_l + I_t \tag{5}$$

比较式(4)和(5)得

$$w_t = -\mathbf{A}_t^{-1}\mathbf{A}_l = [w_{ij}] \tag{6}$$

$$I_t = \mathbf{A}_t^{-1}\Phi \tag{7}$$

式中 $w_t$ 表示了从独立变量神经网络的状态反馈到非独立神经网络的连接矩阵, 为 $(n-1) \times (m-n+1)$ 阶矩阵;  $I_t$ 表示了非独立变量神经网络的偏置, 为 $(n-1) \times 1$ 阶矩阵。

非独立变量神经网络的状态可由独立变量神经网络的状态按式(5)进行简单组合得到。

### 3.2 独立变量 Hopfield 神经网络

只有独立变量神经网络的运动方程才取决于最短路径问题, 它的状态能引导整个神经网络趋于最短路径的解。

定义 Lyapunov 能量函数  $E$ , 它的最小化过程对应神经网络的能量最小化。神经网络的稳定状态对应最短路径的解。

$$E = \sum_{i=1}^m \rho_1 f(v_i) + \rho_2 c_i v_i \tag{8}$$

其中  $f(x) = x^2(x-1)^2$ , 第1项保证了 $v_i \in \{0, 1\}$ 时 ( $i=1,2,\dots,m$ ) 最小, 第2项在所求路径最短时最小。

对式(8)求导, 可得

$$\frac{\partial E}{\partial v_i} = \rho_1 f'(v_i) + \rho_1 \sum_{j=1}^{n-1} w_{ji} f'(v_j) + \rho_2 c_i + \rho_2 \sum_{j=1}^{n-1} w_{ij} c_j \tag{9}$$

由著名的负梯度公式得

$$\frac{dv_i}{dt} = -\rho_1 f'(v_i) - \rho_1 \sum_{j=1}^{n-1} w_{ji} f'(v_j) - \rho_2 c_i - \rho_2 \sum_{j=1}^{n-1} w_{ij} c_j, \tag{10}$$

$i = n, \dots, m$

独立变量神经元采用Hopfield模型, 由一个加法器, 一个电容 $C$ 及输出函数组成<sup>[6]</sup>, 其方程为

$$C_i \frac{dv_i}{dt} = \sum_{j=1}^m w_{ij} x_j(t) + I_{ii} \tag{11}$$

比较式(10)和(11)并令 $C_i=1/\rho_1, k=C_i\rho_2=\rho_2/\rho_1$ 可得到, 对于  $i=n, \dots, m, j=1, \dots, m$ .

$$x_j = g(v_j) = f'(v_j) \tag{12}$$

$$w_{ij} = \begin{cases} -1, & j = i \\ 0, & j \neq i, j = n, \dots, m \\ -w_{ji}, & j = 1, 2, \dots, n-1 \end{cases} \quad (13)$$

$$I_{ii} = -kc_i - k \sum_{j=1}^{n-1} w_{ij}c_j \quad (14)$$

上述独立变量神经网络方程式(11)的矩阵形式为

$$C \frac{dv_i}{dt} = w_i x_i - x_i + I_i \quad (15)$$

式中输出方程:

$$x_i = f'(v_i) = 4v_i^3 - 6v_i^2 + 2v_i \quad (16)$$

$$x_i = f'(v_i) = 4v_i^3 - 6v_i^2 + 2v_i \quad (17)$$

非独立变量神经网络反馈到独立变量神经网络的连接权值矩阵:

$$w_i = -w_i^T = A_i^T(A_i^{-1})^T \quad (18)$$

独立变量神经网络的偏置:

$$I_i = -k w_i^T c_i^T - k c_i^T = k w_i c_i^T - k c_i^T \quad (19)$$

### 3.3 性能模拟

双层递归NN的可变参数只有两个。第 1 个参数是电容  $C$ , 其在式(15)中起到时间常数的作用<sup>[7]</sup>。 $C$  越小, 时间常数越小, 状态收敛越快。另一方面,  $C$  过分小时, 不能保证足够时间搜索最优值, 使计算误差增大, 甚至无解。模拟结果表明,  $C$  取 0.02-0.001 为宜。第 2 个参数是比例系数  $k$ , 它影响路径计算的精度。 $k$  越小, 路径长度项在能量函数中的份额越小, 可能增大神经网络计算得到的路径与最短路径的误差。 $k$  越大, 神经网络计算得到的路径越接近最短路径。但过大时会影响神经网络收敛于 0 或 1。在边的长度取值范围为(0, 1) 时,  $k$  取 1.5 - 4.5 为宜。

为检验基于变量独立性的双层递归神经网络(DVNN)的有效性, 将它的性能与Ali-Kamount方法(A-KNN)进行比较。我们用了 4 个图, G(5,8), G(10,16)<sup>[8]</sup>, G(15,27)和G(20,37), 它们分别有 5, 10, 15 和 20 个节点, 8, 16, 27 和 37 条边。为获得精确的比较, 在后 3 个图中分别随机选取 25 个源/终

点对, 用这两种NN进行计算。在G(5,8)中, 遍历所有节点对也不足 25 对, 通过改变边的长度重复计算。

神经网络的微分方程式采用四阶龙格-库塔法求解, 计算时间步长 $\Delta t=10^{-5}$ , 输出收敛精度 $\Delta V=10^{-5}$ , 各边的长度选择在 0 与 1 之间。DVNN的其它参数选择为:  $C=0.0018$ ,  $k=2.5$ 。A-KNN的其它参数选择为:  $\mu_1=950$ ,  $\mu_2=2500$ ,  $\mu_3=2000$ ,  $\mu_4=475$ ,  $\mu_5=5000$ <sup>[1]</sup>。求得有效路径数、有效路径中最短路径数和平均计算时间如表 1 所示。

可见, DVNN 的收敛速度较快, 找到的最短路径数与 A-KNN 相当(准确率相当), 但找到有效路径(不一定最短)的成功率为 96%, 比 A-KNN 的 92%要高, 也就是可靠性高。

提高准确率与变量收敛于 0 或 1 是矛盾的。前者会使神经网络的最终状态可能并不收敛与 0 或 1, 但上述两个因数的作用是相互分离的。最短路径因数仅表现在神经网络的偏置项中, 在DVNN中仅表现在独立变量神经网络的偏置项 $I_i$  中, 因此, DVNN偏置项采取如下措施: 先考虑偏置项 $I_i$ , 使神经网络的状态在最短路径要求下收敛到一定的精度, 然后再将偏置项 $I_i$ 置 0, 从而让状态变量收敛于 0 或 1。即上表中 DVNN的计算时间是两次收敛的时间。而A-KNN计算时间只有一次收敛, 其神经元的最终输出为“1”或“0”要根据其输出是否大于 0.5 而定。

## 4 基于双层递归 NN 的路由算法

对于  $n$  个节点,  $m$  条支路的网络(有向图), 根据 M/M/1 假设, 网络总的平均时延:

$$T = D(f_i) = \sum_{i=1}^m \frac{f_i}{(r_i - f_i)^2} \quad (20)$$

其中 $f_i$ 为链路 $l$ 上的流量;  $r_l$ 为链路 $l$ 的容量。

在 FD 中, 链路的成本(或边的长度)取决于链路时延的一阶导数。在每次的循环运算中, 流量的调整使成本变为时变的, 都要重新计算 SP, 因此在实时选择路由的应用中, 计算 SP 有严格的时间限制。NN 是很好的选择, 用 NN 计算 SP 与 FD 相结合, 可使优化计算速度达到实时应用的要求。算法中各参数的定义如下:  $W$  为所有通信的源/终点对的

表 1 两种神经网络的路径计算模拟结果

有向图 G(n,m)	G(5,8)	G(10,16)	G(15,27)	G(20,37)	
A-KNN	有效路径数	23	25	23	21
	最短路径数	23	25	23	21
	平均计算时间(ms)	20.0512	21.332	25.266	43.106
DVNN	有效路径数	25	25	23	23
	最短路径数	25	24	22	21
	平均计算时间(ms)	10.1076	13.6844	16.492	16.2428

集合;  $d_w$  为源/终点对之间  $w \in W$  的通流量;  $p_w$  为源/终点对之间所有有向路径的集合;  $f_p$  为路径  $p$  的流量;

算法描述如下:

(1)初始化 根据各源/终点对之间的业务通流量  $d_w$ , 分别对源/终点对之间各条路径分配流量, 使  $\sum_{p \in p_w} f_p = d_w, f_p \geq 0,$

$\forall p \in p_w, w \in W.$

(2)在第  $k$  次循环, 计算平均时延对各链路的一阶导数作为链路成本,

$$c_l^k = D'(f_l^k) = \frac{r_l}{(r_l - f_l^k)^2} \quad (21)$$

其中

$$f_l^k = \sum_{p \in p_w} f_p^k \zeta(p, l) \quad (22)$$

$$\zeta(p, l) = \begin{cases} 1, & \text{若路径 } p \text{ 包含链路 } l \\ 0, & \text{其它} \end{cases}$$

(3)每个源/终点对之间用 DVNN 计算最小一阶导数长度路径(MFDL), 将它们间的通流量  $d_w$  全部投放在 MFDL 上, 形成路径流量  $\bar{f}_p^k$ 。

(4)更新路径流量:

$$f_p^{k+1} = f_p^k + \alpha_k (\bar{f}_p^k - f_p^k) \quad (23)$$

选择  $\alpha_k$  使  $D[f_p^k + \alpha_k (\bar{f}_p^k - f_p^k)] = \min_{\alpha \in [0,1]} D[f_p^k + \alpha (\bar{f}_p^k - f_p^k)]$ ,

$\alpha \in [0, 1]$ 。

(5)计算  $|T^{k+1} - T^k|$ , 若小于所设精度, 则终止; 否则, 返回(2)再循环。

利用上述 NN 与 FD 相结合的算法, 对文献[1]的 5 个节点的网络进行了仿真, 在优化过程中, 所优化的目标函数-总平均时延的变化曲线如图 1 所示。可见该算法能够稳定收敛。

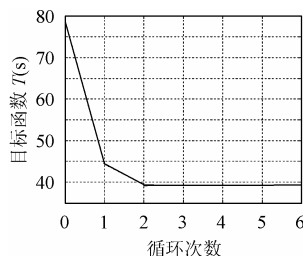


图 1 平均时延优化过程波形

## 5 结束语

本文介绍了由独立变量神经元和非独立变量神经元组成的双层递归神经网络, 用于计算最短路径, 它具有神经元数目少、运算速度快、寻找路径的成功率高等特点。

将该神经网络与 FD 路由算法相结合, 可以提高路由优化速度, 达到实时处理的要求。

## 参考文献

- [1] Ali M Kamount F. Neural networks for shortest path computation and routing in computer networks. *IEEE Trans. on Neural Networks*, 1993,4(6): 941 – 953.
- [2] Fratta L, Gerla M and Kleinrock L. The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 1973, 3(1): 97 – 133.
- [3] Schwartz M and Cheung C. The gradient projection algorithm for multiple routing in message switched systems. *IEEE Trans. on Communications*. 1976, Com-24 (4): 449 – 456.
- [4] Filip A, Bernardete R, Luis R. A neural network for shortest path computation. *IEEE Trans. on Neural Networks*, 2001, 12(5): 1067 – 1073.
- [5] Strum R D, Ward J R. *Electric Circuits and Networks* (2<sup>nd</sup> edition). Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1985, 345 – 358.
- [6] Hopfield J J and Tank D W. Neural computation of decisions in optimization problems. *Biolog. Cybern.*, 1985, 52(3): 141 – 152.
- [7] Wang J. A recurrent neural network for solving the shortest path problem. *IEEE Trans. on Circuits and Systems \_ I: Fundamental Theory and Applications*, 1996, 43(6): 482 – 486.
- [8] Bu L and Chiueh T. Solving the shortest path problem using an analog network. *IEEE Trans. on Circuits and Systems \_ I: Fundamental Theory and Applications*, 1999, 46(11): 1360 – 1363.

周井泉: 男, 1963 年生, 副教授, 博士生, 研究方向为通信网络, 神经网络与智能优化。

张顺颐: 男, 1944 年生, 教授, 博士生导师, 研究方向为计算机通信, IP 网络技术以及管理信息系统。