基于 BP 算法的加权模糊 Petri 网权值学习算法

吴荣海,范晓梅
(大理学院 数学与计算机学院,云南 大理 671003)

摘 要: 加权模糊 Petri 网缺乏较强的自学习能力,针对这个问题,给出了一个基于 BP 算法的加权模糊 Petri 网权值学习算法。该算法不需要对原有模型进行修改,使得加权模糊 Petri 网权值的学习和训练得到一定地简化。

关键词: WFPN;产生式规则;BP 算法;权值学习
中图分类号: TP18　　**文献标识码**: A　　**文章编号**: 1674-7720(2012)13-0068-02

Weight learning algorithm of weighted fuzzy Petri net based on BP algorithm

Wu Ronghai, Fan Xiaomei
(School of Mathematics and Computer Sciences, Dali University, Dali 671003, China)

Abstract: Weighted Fuzzy Petri Nets (WFPN) lack strong self-learning abilities, to address the issue, a weighted learning algorithm of WFPN based on BP algorithm is presented. According to the algorithm, without modifying the original WFPN model, simplify the process of learning and training of WFPN.

Key words: Weighted Fuzzy Petri Net(WFPN); production rule; BP(Back Propagation) algorithm; weight learning

加权模糊 Petri 网 WFPN(Weighted Fuzzy Petri Net)为由加权模糊产生式规则所构成的知识库系统建模提供了的良好工具,它能够将规则系统中的知识结构化地表示出来。但自适应能力差是模糊系统本身的一个不足之处,加权模糊产生式规则中的部分参数(例如命题权值、规则的确信度等),这些参数往往依赖于领域专家的经验,很难精确地获得,影响了 WFPN 的知识推理[1]。在参考文献[1-5]中,研究人员对模糊 Petri 的学习能力做了进一步研究并给出了多个模型以及对应的学习算法。

WFPN 中的变迁与库所之间的连接有着明确的意义,表示了各个命题之间的蕴涵关系,这是与一般的人工神经网络不同的地方[1]。WFPN 中人工神经网络中的层次结构不明显,将 BP 算法引入 WFPN 中需要对 BP 算法做一些修改,本文在参考文献[6]给出的 WFPN 模型以及相应的推理算法的基础上,将 BP 算法应用在不存在回路的 WFPN 模型中,对 WFPN 模型中的权值进行学习、优化,使其接近理想值,从而提高模型的自适应能力,文中所给算法不需要通过增加虚变迁和虚库所[1]对 WFPN 模型进行层次划分,这样可以避免增加 WFPN 模型的复杂度。

1 WFPN 模型

参考文献[6]给出了 WFPN 的一般形式及推理算法。WFPN 为一个十元组 (P,T,D,I,O,M,Th,W,f,β),基于该 WFPN 模型的推理算法采用了矩阵运算。WFPN 的一般形式与推理算法可参考文献[6]。

2 WFPN 的命题权值学习和训练

对于多输入多输出的模糊产生式规则可以等价地分解为多个多输入单输出的模糊产生式规则。因此,本文仅针对多输入单输出产生式规则系统进行讨论,为了便于后续讨论,先给出关于目标库所的定义[7]。

定义 1 目标库所:加权模糊 Petri 网中,如果存在库所 $p_i \in P$,满足 $\forall t \in T, p_i \notin I(t)$,那么库所 p_i 称为加权模糊 Petri 网的目标库所,和这些库所关联的命题的可信度完全依赖推理得到,是知识库最终的推理结果。

2.1 WFPN 中的反向递推(BP)算法

在 WFPN 中以每个变迁为中心,用误差反传算法来调整变迁输入弧上的权值。设 WFPN 模型有 r 个目标库所 $p_j\ j=1,2,\cdots,r$。用 k 批样本数据进行学习。取误差代价函数 E 为[1]:

$$E=\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{r}(M_i^d(p_j)-M_i(p_j))^2 \qquad (1)$$

式(1)中 $M_i^d(p_j)$、$M_i(p_j)$分别表示目标库所 p_j 的第 i 批样本数据的期望标记值和实际标记值。以下分别计算输出库所为目标库所的变迁及输出库所不是目标库所的变迁的一阶梯度。

(1)计算输出库所为目标库所的变迁 t_i 的一阶梯度

若 $\exists t_i\in T$,t_i 的输入弧上的权值分别是:$w_{1i}, w_{2i}, \cdots, w_{ji}, \cdots, w_{|I(t_i)|i}$,其中 w_{ji} 是从变迁 t_i 的输入库所 p_j 到变迁 t_i 的输入弧上的权值。若 $\exists p_g\in P\wedge p_g\in O(t_i)\wedge O(p_g)=\Phi$,则库所 p_g 是变迁 t_i 的输出库所且库所 p_g 是网中的目标库所,利用误差反传算法计算一阶梯度,由式(2)计算:

$$\frac{\partial E}{\partial w_{ji}}=\frac{\partial E}{\partial M(p_g)}\frac{\partial M(p_g)}{\partial w_{ji}}=(M^d(p_g)-M(p_g))\frac{\partial M(p_g)}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial w_{ji}}=\begin{cases}\delta_g\beta_i M(p_j), & N(j,i)=0\\ \delta_g\beta_i(1-M(p_j)), & N(j,i)=1\end{cases} \qquad (2)$$

式(2)中 E 是误差代价函数,$M(p_j)$是库所 p_j 的标记值,w_{ji} 是从库所 p_j 到变迁 t_i 的输入弧上的权值,δ_g 与 β_i 分别计算如下:

$$\delta_g=M^d(p_g)-M(p_g) \qquad (3)$$

$$\beta_i=\begin{cases}f(t_i), & if(|I(p_g)|=1\wedge p_g\in O(t_i)\wedge x_i\geqslant Th(t_i))\\ f(t_i)/\sum_{k=1}^{|I(p_g)|}f(t_k), & if(|I(p_g)|>1\wedge p_g\in O(t_i)\wedge x_i\geqslant Th(t_i))\\ 0, & x_i<Th(t_i)\end{cases} \qquad (4)$$

(2)计算输出库所不是目标库所的变迁 t_i 的一阶梯度

若 $\exists t_i\in T$,t_i 的输入弧上的权值分别是:$w_{1i}, w_{2i}, \cdots, w_{ji}, \cdots, w_{|I(t_i)|i}$,其中 w_{ji} 是从变迁 t_i 的输入库所 p_j 到变迁 t_i 的输入弧上的权值。若 $\exists p_n\in P\wedge p_n\in O(t_i)\wedge O(p_n)\neq\Phi$,则库所 p_n 是变迁 t_i 的输出库所且库所 p_n 不是网中的目标库所,利用误差反传算法计算一阶梯度得:

$$\frac{\partial E}{\partial w_{ji}}=\frac{\partial E\partial M(p_n)}{\partial M(p_n)\partial w_{ji}}=\begin{cases}\beta_i M(p_j)\sum_{m=1}^{|O(p_n)|}\beta_m r w_{nm}\frac{\partial E}{\partial M(p_m)}, & N(j,i)=0\\ \beta_i(1-M(p_j))\sum_{m=1}^{|O(p_n)|}\beta_m r w_{nm}\frac{\partial E}{\partial M(p_m)}, & N(j,i)=1\end{cases} \qquad (5)$$

其中,$r=\begin{cases}1, & N(n,m)=0\\ -1, & N(n,m)=1\end{cases}$

式(5)中 $\frac{\partial E}{\partial M(p_n)}$ 的计算需要注意,因为库所 p_n 不是网中的目标库所,所以在计算 $\frac{\partial E}{\partial M(p_n)}$ 时要考虑到库所 p_n 与它的所有输出变迁的连接,设:$t_m\in O(p_m)\wedge t_m\in T\wedge p_m\in P\wedge p_m\in O(t_m)$,具体算法如下:

$$\frac{\partial E}{\partial M(p_n)}=\sum_{m=1}^{|O(p_n)|}\frac{\partial E\partial M(p_m)}{\partial M(p_m)\partial M(p_n)}=\sum_{m=1}^{|O(p_n)|}\frac{\partial E\partial M(p_m)\partial y_m\partial x_m}{\partial M(p_m)\partial y_m\partial x_m\partial M(p_n)} \qquad (6)$$

式(6)中 $|O(p_n)|$是库所 p_n 输出变迁的个数,p_m 是库所 p_n 的输出变迁 t_m 的输出库所,$M(p_m)$是库所 p_m 的标记值。

通过式(2)和式(6)进行反向递推,计算出网中各个变迁输入库所对应的输入弧上的权值改变时,误差 E 的导数后,给出网中各个变迁 t_i 的输入弧上权值的学习算法,其中 η 为学习率。

$$w_{ji}(k+1)=w_{ji}(k)+\eta\frac{\partial E}{\partial w_{ji}} \qquad (7)$$

为了保证变迁 t_i 的输入弧上的权值总和为 1,在利用式(7)修改变迁 t_i 的输入弧上的权值后,令:

$$w_{ji}=w_{ji}/\sum_{l=1}^{|I(t_i)|}w_{li} \qquad (8)$$

式(8)中,w_{li} 是从变迁 t_i 的输入库所 p_l 到变迁 t_i 的输入弧上的权值。

2.2 WFPN 的命题权值学习和训练的步骤

(1)对需要学习的参数赋予初值。

(2)对 r 批样本数据,执行 WFPN 的正向推理算法。

(3)计算误差代价函数 E,若 $E<\varepsilon$,结束学习;否则,继续下一步。

(4)执行 BP 算法,对参数进行学习,返回步骤(2)继续。

在 WFPN 模型上应用 BP 算法时,同样有收敛速度慢的问题,为了加快算法收敛速度,训练过程中引入了变学习率[8]的方法。

$$\eta=\begin{cases}1.05\eta & E(n)<E(n-1)\\ 0.7\eta & E(n)\geqslant 1.04E(n-1)\\ \eta, & \text{其他}\end{cases} \qquad (9)$$

在学习过程中,根据 $E(n)$与 $E(n-1)$之间的大小关系,按照式(9)动态修改学习率 η。

2.3 实例计算

实验引用参考文献[1]的基于模糊产生式规则的专家系统,其对应的 FPN 模型如图 1 所示。

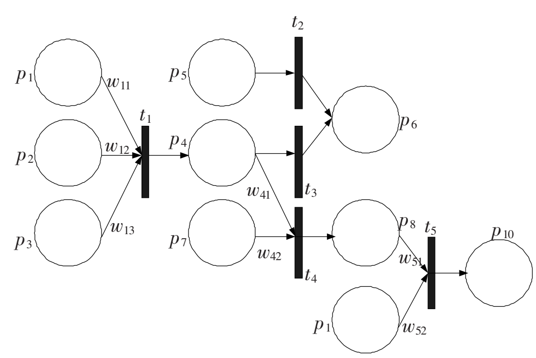

图 1 加权模糊 Petri 网模型

假设图 1 中 WFPN 各个权值参数的理想值是:$w_{11}=0.3$,$w_{12}=0.5$,$w_{13}=0.2$,$w_{41}=0.62$,$w_{42}=0.38$,$w_{51}=$

0.35，w_{52}=0.65。

用 90 批样本数据对图 1 所示的 WFPN 模型进行训练，初始参数为：

w_{11}=0.1，w_{12}=0.1，w_{13}=0.8，w_{41}=0.2，w_{42}=0.8，w_{51}=0.8，w_{52}=0.2。

WFPN 模型经过 41 次的学习，参数调整为：

w_{11}=0.299 993，w_{12}=0.500 043，w_{13}=0.199 965，w_{41}=0.619 675，w_{42}=0.380 325，w_{51}=0.349 974，w_{52}=0.650 026。

用 10 组非样本数据中的输入数据对训练后的 WFPN 进行模糊推理，结果见表 1。

表 1　实际输出和期望输出

序号	p_6		p_{10}	
	实际输出	期望输出	实际输出	期望输出
1	0.539 037	0.539 032	0.492 728	0.492 725
2	0.560 501	0.560 505	0.597 822	0.597 814
3	0.586 434	0.586 431	0.520 070	0.520 055
4	0.686 626	0.686 624	0.622 721	0.622 689
5	0.671 761	0.671 763	0.550 355	0.550 327
6	0.492 619	0.492 612	0.424 258	0.424 270
7	0.570 969	0.570 971	0.514 670	0.514 645
8	0.669 675	0.669 676	0.309 258	0.309 284
9	0.647 340	0.647 342	0.480 594	0.480 608
10	0.522 189	0.522 194	0.524 476	0.524 457

本文针对没有回路的 WFPN 模型，提出了 WFPN 模型的学习算法。学习算法是借鉴神经网络中的 BP 算法，但该学习算法是直接建立在 WFPN 模型上的，不需要将 WFPN 转化到神经网络模型上，使得 WFPN 具有像神经网络中 BP 网络一样的学习能力。

参考文献

[1] 鲍培明.基于 BP 网络的模糊 Petri 网的学习能力[J].计算机学报，2004，27(5)：695-702.

[2] Li Xiaoou，ROSANO L F. A weighted fuzzy petri net model for knowledge learning and reasoning. Neural Networks[R]. 1999. IJCNN'99. International Joint Conference on Volume 4，July 1999.

[3] Li Xiaoou，Yu Wen，ROSANO L F. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework [J]. Systems，Man and Cybernetics，Part C，IEEE Transactions on Volume 30，2000，11(4)：442-450.

[4] TSANG E C C，YEUNG D S，LEE J W T. Learning capability in fuzzy Petri nets. Systems [R]. Man，and Cybernetics，1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on Volume 3，1999.

[5] 吴宴华，须文波.结合遗传算法优化模糊 Petri 网的参数[J].微计算机信息，2005，21(12-2)：174-175.

[6] 吴荣海，范晓梅，吴坚，等.加权模糊 Petri 网的正向推理算法[J].大理学院学报，2007，6(8)：68-72.

[7] 吴荣海.加权模糊 Petri 网在不精确知识表示和推理中的应用研究[D].云南：云南师范大学，2006.

[8] 史忠植.知识发现[M].北京：清华大学出版社，2004，230-264.

(收稿日期：2012-02-13)

作者简介：

吴荣海，男，1975 年生，硕士，讲师，主要研究方向：Petri 网应用，计算机应用。