# 基于结果反馈的模糊 Petri 网学习算法①

严军辉[1]　方路平[2]　肖寒冰[1]　魏渊洁[1]　谢超[1]　(1. 浙江工业大学 计算机科学与技术学院 浙江 杭州 310023; 2. 浙江工业大学 信息工程学院 浙江 杭州 310023)

**摘　要：** 针对模糊 Petri 网模型的复杂结构，在不增加虚库所和虚变迁的情况下改进了模糊 Petri 网分层算法，从而简化模糊 Petri 网学习和训练方法。为提高收敛速率，本文从一个全新的角度考虑模糊 Petri 网的学习和训练，提出了基于结果反馈的模糊 Petri 网学习的新算法(FBFPN)。该算法通过对纯网进行层次式分层及建立变迁点燃的近似连续函数后，调整权值、变迁的阈值、变迁的可信度的同时又调整输入矢量的多重作用来最小化误差函数。仿真结果分析表明，该算法具有良好的学习效率和泛化能力。

**关键词：** 模糊 Petri 网；反向传播；结果反馈；库所；变迁

## Learning Algorithm of Fuzzy Petri Net Based on Result-Feedback

YAN Jun-Hui[1], FANG Lu-Ping[2], XIAO Han-Bing[1], WEI Yuan-Jie[1], Xie Chao[1] (1. College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China; 2. College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** With regard to the complex structure of Fuzzy Petri Net, this paper improved the hierarchical algorithm of Fuzzy Petri Net without increasing the virtual place and virtual transition, thereby simplifying the learing and training of Fuzzy Petri Net. To speed convergence, this paper proposed a new algorithm for the learning of Fuzzy Petri Net based on the results feedback, namely FBFPN. Firstly, this algorithm layered the pure net hierarchically and established the approximate continuous function of the transition firing, then adjusted the weight, the threshold and the credibility ,finally adjusted the input vector to minimize the error function. Simulation results showed that this algorithm has stronger generalization ability and higher learning efficiency.

**Keywords:** FPN; BP; Result-Feedback; place; transition

模糊 Petri 网是基于模糊产生式规则知识库系统的良好的建模工具[1]，可将知识的表达与推理融为一体，其中的一些参数，例如，权值、阈值、确信度等，它们通过人们长期经验获得，但是难以精确获得，甚至根本不可能获得，这阻碍了模糊 Petri 网的推理。由于模糊 Petri 网自学习调整能力差是其固有的缺点[2,3]。Looney 提出了具有学习能力的模糊 Petri 网，实现对阈值的调整。随后又有研究者将模糊 Petri 网结合神经网络学习算法研究了模糊 Petri 网的学习能力，通过修改传统的 BP 算法实现前向推理和反向权值调整。研究者 X. Li 等人提出了较严格条件下的权值学习问题，在此基础上提出一种自适应模糊 Petri 网 AFP(Adaptive Fuzzy Petri Nets)。由于传统的 BP 算法是基于梯度下降的误差反向传播学习算法，收敛速度慢且易于陷入局部最优解。汤新民等提出模糊 Petri 网元模型将模糊产生式规则中的四种基本规则对应的

① 基金项目:浙江省财政厅专项(2008C0417)
收稿时间:2010-04-15;收到修改稿时间:2010-05-13

模糊 Petri 网统一起来，最终采用 Levenberg -Marquardt 算法实现权值优化[4,5]。

以上方法都对误差反向传播算法的性能进行了一些改善。但总的来讲，主要都从权值调整的角度去考虑。本文从一个新的角度考虑 Petri 网的学习训练，结合结果反馈算法、传统 BP 与模糊 Petri 网，提出了基于结果反馈的 FPN 算法(记做 FBFPN 算法)。FBFPN 的训练可以看作是输入矢量到输出矢量的非线性映射，记作 $O=F(W;\ \gamma;\ \beta;X)$，其中 O 表示输出矢量，W 表示权值，γ 表示变迁的阈值，β 表示变迁的可信度，X 表示输入矢量。可以看到，除了能调整权值 W，变迁的阈值 γ，变迁的可信度 β，还可以调整输入矢量 X。FBFPN 算法将模糊 Petri 网输入调整与通常的权值的反向传播算法结合起来，更有效地提高 FPN 的学习性能。

## 1 FPN模型的建立

### 1.1 FPN 定义

首先基于模糊产生式规则上建立 FPN 的定义[2]：

定义 1. 一个 FPN 为一个九元组，FPN={P, T, F, I, O, f, M, S, W}，其中：

$P=\{p_1,\cdots p_i,\cdots p_n\}$，表示库所结点的有限集合，反映了模糊命题。

$T=\{t_1,\cdots t_i,\cdots t_n\}$，表示变迁结点的有限集合，反映了规则的实现。

$F\in(P\times T)\cup(T\times P)$，表示库所到变迁或变迁到库所的有限集合。

I(O):T→P，是输入(输出)函数，表示变迁到库所的映射。

f：T→[0,1]，是一个映射，赋予变迁的可信度 $f(t)=\beta$。

M：T→[0,1]，是一个映射，对变迁结点 $t(t\in T)$ 定义一个阈值 $M(t)=\gamma$。

S：P→[0,1]，是一个映射，每一个库所 $pi(pi\in P)$ 有一个可信度值 S(pi)，反映命题的真实程度。

$W=\{w_1,\cdots w_i,\cdots,w_r\}$，是库所的权值集合,表示变迁的输入库所所占的权重，反映了规则中前提条件对结论的支持程度。

### 1.2 FPN 分层算法的改进

现实世界中，在许多情况下是由因到果，而不能由果再推到因，反映到模糊 Petri 网中就是 FPN 模型中没有回路，也就是模糊 Petri 网是纯网。并且有回路的 FPN 会导致 FPN 模型的不稳定。如图 1 所示，假设初始状态：库所 $p_1$ 的可信度为 0.9；变迁 $t_1$ 的可信度为 0.9，阈值为 0.2；变迁 $t_2$ 的可信度为 0.9，阈值为 0.2。通过推理点燃变迁 $t_1$ 可以得出库所 p2 的可信度为 0.81，然后通过推理点燃变迁 $t_2$ 又可以得出库所 $p_1$ 的可信度为 0.729，如此反复循环，库所 $p_1$ 与库所 $p_2$ 的可信度在不断改变，导致模型的不稳定。所以一般情况下模糊 Petri 网是没有回路，也就是模糊 Petri 网是纯网。

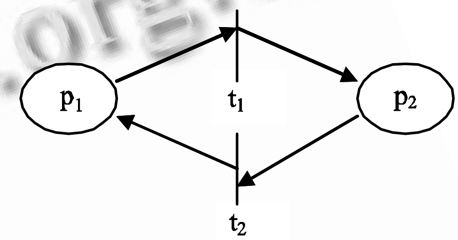


图 1 有回路的 FPN

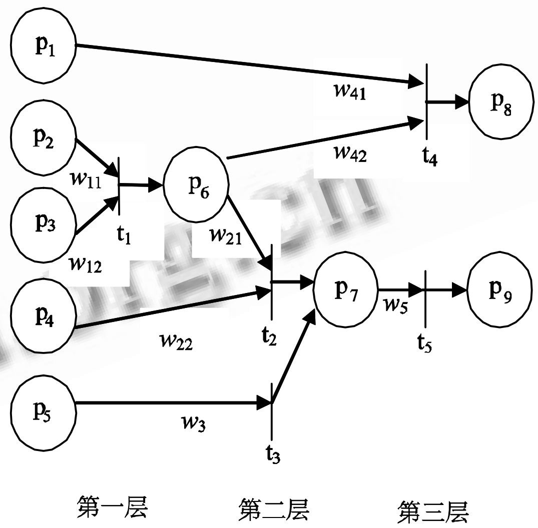


图 2 分层后的 FPN 模型

FBFPN 算法将神经网络输入调整与通常的权值的反向传播算法结合起来，所以首先要对无回路的 FPN 进行分层。文献[2]通过添加虚库所和虚变迁对复杂 FPN 模型进行分层，这增加了 FPN 模型的结构复杂度。本文在不添加虚库所和虚变迁的前提下，对 FPN 模型进行分层，从而减少了模型的结构复杂度。改进的分层算法如下，图 2 表示分层后的 FPN 模型。

Step1：初始化。建立库所集合 PM、PN，PM={p| p 为初始库所}，PN={p| p 为终止库所}；设 i=1，i 是循环变量，i 标记模型的分层数。

Step2：建立变迁集合 $T_i=\{t\in T\mid \forall p\in I(t), p\in PM;$ 且 $\forall p\notin O(t), p\in PN\}$ 在 T 中寻找变迁 t，t 的所有输入库所都属于 PM，并且 t 的所有输出库所都不属于 PN，这样的 t 构成集合 $T_i$，$T_i$ 表示同一层中变迁的集合。

Step3：若 $\exists t\in T_i$，$\exists p\in O(t)$；且 $\exists t'\in T-T_i$，$p\in O(t')$；则 $T_i=T_i-t$。

这一步反复迭代，目的是：若果一个库所是多个变迁的输出库所，则这些变迁分在同一个层中。

Step4：$PM=PM\cup\{p\mid p\in O(t),\text{且}\forall t\in T_i\}$。

$T_i$ 中所有变迁的输出库所都插入到 PM 中。

Step5：若 P=PM+PN，执行 Step6。否则，i=i+1，返回 Step2。

Step6：$T_{i+1}=T-\{T_1+\cdots+T_i\}$。这是保证最后一层是输出层。

### 1.3 FPN 推理中连续函数的建立

FPN 既为模糊产生式规则建立了一个直观的、图形化的模型，又为模糊推理建立了结构化的推理机制。而基于形式化推理得出的结果不是一个连续函数，对一阶求导带来问题。因而，必须建立一个连续的激发函数以便求导。参照文献[6]中的一个 S 形函数，建立变迁点燃连续函数和最大(max)、最小(min)运算的连续函数。这个函数适合 FBFPN 的近似推理机制，又适用于 FBFPN 的学习过程。

设 y(x)是一个 S 形函数，b 是一个常量，

$$y(x)=1/(1+e^{-b(x-k)}) \tag{1}$$

当 b 足够大时，(a)若 x<k 时 $e^{-b(x-k)}\to\infty$，则 $y(x)\approx 0$。(b)若 x>k 时，$e^{-b(x-k)}\approx 0$，则 $y(x)\approx 1$。y(x)函数在本文中有两处应用。

1) 建立最大运算连续函数

利用 y(x)函数，当 b 足够大时，显然下式推导正确：

$$t=\max(x_1,x_2)\approx x_1/(1+e^{-b(x_1-x_2)})+x_2/(1+e^{-b(x_2-x_1)}) \tag{2}$$

$$\begin{aligned} h&=\max(x_1,x_2,x_3)=\max(\max(x_1,x_2),x_3)=\max(t,x_3)\\ &\approx t/(1+e^{-b(t-x_3)})+x_3/(1+e^{-b(x_3-t)}) \end{aligned} \tag{3}$$

2) 判断变迁的使能，建立变迁点燃连续函数

设 $x=\sum_{j=1}^{n}S(p_j)\times w_j$，k=M(t)，则 y(x)函数建立了变迁的使能判断。当 b 足够大时，(a)若 x<k 时，即 $\sum_{j=1}^{n}S(p_j)\times w_j<M(t)$ 时，$e^{-b(x-k)}\to\infty$，则 $y(x)\approx 0$，表示变迁 t 没有点燃，对输出库所传送的为值 0。(b)若 x>k 时，即 $\sum_{j=1}^{n}S(p_j)\times w_j>M(t)$ 时，$e^{-b(x-k)}\approx 0$，则 $y(x)\approx 1$，表示变迁 t 使能。对输出库所传送的值为 $y(x)\times f(t)\times\sum_{j=1}^{n}S(p_j)\times w_j$。

## 2 FPN的反向传播学习和网络反转

### 2.1 FPN 的反向传播学习

FPN 的反向传播学习主要是借鉴 BP 网络。FPN 中的第一层称为输入层，最后一层称为输出层。设 FPN 具有 m 层，各个库所的输入输出关系函数是 f(x)，由第 k-1 层的第 i 个库所 $s_i^{k-1}$ 到第 k 层的第 j 个变迁 $t_j^k$ 的连接权值为 $w_{ij}$，并设第 j 个变迁 $t_j^k$ 输入总和为 $u_j^k$ 且输出为 $s_j^k$，则各变量之间的关系为

$$u_j^k=\sum_{i=0}^{n}w_{ij}s_i^{k-1}; \tag{4}$$

$$s_j^k=f(u_j^k) \tag{5}$$

k=1，…，m

其中，$f(x)=bx/(1+e^{-b(x-g)})$ (6)

β 表示该变迁的可信度，γ 表示该变迁的阈值。

并设目标函数为：$J=\frac{1}{2}\sum_{j=1}^{n}(s_j^m-s_j)^2$ (7)

反向传播学习算法一阶梯度的计算可参照文献[7].

权值调整为：$w_{ij}=w_{ij}+\Delta w_{ij}$ (8)

其中，$\Delta w_{ij}=-h\frac{\partial J}{\partial w_{ij}}=-h\frac{\partial J}{\partial u_j^k}\frac{\partial u_j^k}{\partial w_{ij}}=-hd_j^k s_i^{k-1}$ (9)

$$d_j^k=\frac{\partial J}{\partial u_j^k}=\frac{\partial J}{\partial s_j^k}\frac{\partial s_j^k}{\partial u_j^k}=\frac{\partial J}{\partial s_j^k}f'(u_j^k)$$

$$=\begin{cases} f'(u_j^m)(s_j^m-s_j)\text{输出层的权值调整}\\ f'(u_j^k)\sum_l w_{lj}d_l^{k+1}\text{非输出层的权值调整} \end{cases} \tag{10}$$

其中，$f'(x)=\frac{f(x)}{x}+bf(x)-\frac{bf^2(x)}{bx}$ (11)

η 称为学习率。

同理对变迁可信度的调整为：$b_i=b_i+\Delta b_i$ (12)

同理对于变迁阈值的调整为：$g_i=g_i+\Delta g_i$ (13)

为了克服梯度下降法的不足，主要是收敛速度慢，也可进行一些优化方法。比如：变步长法、引入动量项、采用 L—M 算法等。

### 2.2 网络反转在 FPN 的应用

网络反转是指根据神经网络输出的矢量的误差修改输入矢量，以达到获得理想的输出矢量的目的[8]。本文对 FPN 模型分层后，采用网络反转思想，且对 FPN 的输入矢量调整也采用梯度下降法：

$$s_i = s_i + \Delta s_i \tag{14}$$

下面推导输入矢量的调整算法：

对于 m 层上的变迁 $t_j^m$ 的输入矢量 $s_i^{m-1}$ 一阶梯度为：

$$\Delta s_i^{m-1} = -h\frac{\partial J}{\partial s_i^{m-1}} = -h\frac{\partial J}{\partial u_j^m}\frac{\partial u_j^m}{\partial s_i^{m-1}} = -hc_j^m\frac{\partial u_j^m}{\partial s_i^{m-1}} \tag{15}$$

同理 m-1 层上的变迁的输入矢量一阶梯度为：

$$\Delta s_i^{m-2} = -h\frac{\partial J}{\partial s_i^{m-2}} = -h\frac{\partial J}{\partial u_j^{m-1}}\frac{\partial u_j^{m-1}}{\partial s_i^{m-2}} = -hc_j^{m-1}\frac{\partial u_j^{m-1}}{\partial s_i^{m-2}} \tag{16}$$

依次类推，继续反向递推，依次对 m-2，…，1 层，求得所需的输入矢量的一阶梯度。

传统的 FPN 参数训练只局限于权值训练，而本文通过网络反转思想对输入矢量进行微小的调整，等效于一个很小的模糊化，从而提高训练速度。

## 3 基于结果反馈的模糊Petri网的新算法

本文提出基于结果反馈的模糊 Petri 网的新算法(FBFPN)，调整权值、变迁的阈值、变迁的可信度的同时又调整输入矢量的多重作用来最小化误差函数。其算法流程图如图 3 所示。FBFPN 算法如下：

**Step1**：首先对 FPN 进行分层，依次记为第 1 层，第 2 层，…，第 m 层。

**Step2**：对库所到变迁的权值、变迁的阈值、变迁的可信度，赋初值；并给出训练允许误差 e 和最大学习次数 p。

**Step3**：输入 r 批样本，依次计算各层变迁的输出值。

**Step4**：计算学习次数 n，若 n>p，则转 Step8，否则下一步。

**Step5**：计算目标函数 J，若 J<e，则转 Step8，否则下一步。

**Step6**：求各个参数的误差梯度 $\partial J/\partial w_{ij}$、$\partial J/\partial g_i$、$\partial J/\partial b_i$、$\partial J/\partial s_i$，计算 $\Delta w_{ij}$、$\Delta g_i$、$\Delta b_i$、$\Delta s_i$。

**Step7**：调整权值、变迁可信度、变迁阈值、输入矢量，转 Step3。

**Step8**：推理结束，得到权值、变迁可信度、变迁阈值。

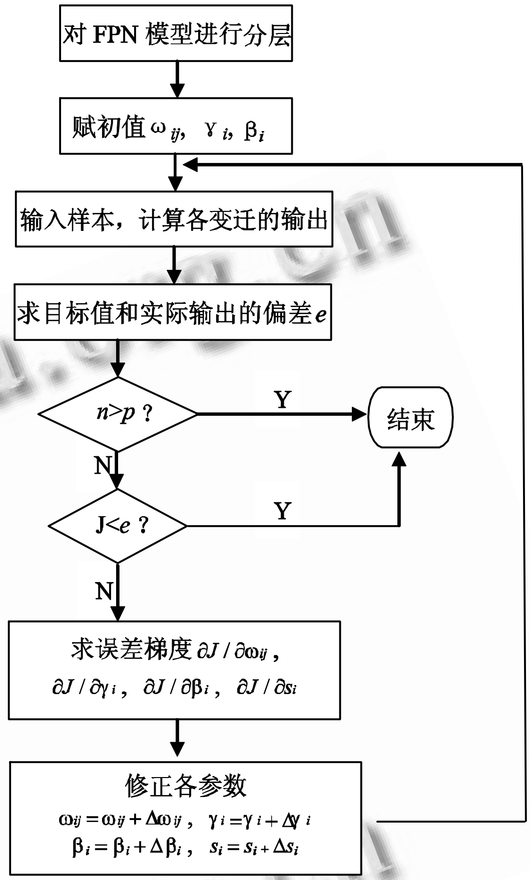


图 3 FBFPN 参数优化的算法程序框图

## 4 FBFPN实验及性能分析

训练算法的好坏主要用两个性能来评价。一是学习效率，二是泛化能力。为了验证本文提出的 FBFPN 算法性能，对图 1 中的模型进行实验分析，并与 FPN 梯度下降算法做比较。

假设理想的参数设置如下 W= [0.4, 0.6, 0.3, 0.7, 0.25, 0.7, 0.3, 0.75]，变迁的可信度 $\beta$ =[0.92, 0.95, 0.93, 0.89]，变迁的阈值 $\gamma$ =[0.2, 0.25, 0.23, 0.22]。给定初始权值向量为 W=[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]，$\beta$ =[0.9, 0.9, 0.9, 0.9] $\gamma$ =[0.2, 0.2, 0.2, 0.2]，采用 FBFPN 算法，取 $\eta$ =0.5，训练样本数为 50 组数据，再用 50 组数据作为测试样本数，训练次数定为 500 次。学习的结果如表 1 所示。表 2 表示输入矢量训练后的数据，括号内表示初始值。从表 2 来看，有的数据变小了，有的数据变大了，但

是变化都很小，不过通过这种思想的训练，却带来性能很大的提高。

从表 1 中可得出，在训练误差上，FBFPN 算法比 FPN 梯度下降算法减少了很多，也就是说 FBFPN 学习效率高了很多；从测试误差或泛化能力角度来看，FBFPN 和 FPN 梯度下降算法差不多。总的来讲 FBFPN 训练效率比梯度下降法要好很多。

表 1　学习和测试结果(学习次数 500)

| 学习算法 | 训练误差 | 测试误差 |
|---|---|---|
| FPN 梯度下降算法 | 0.0081095771 | 0.0922794468 |
| FBFPN 算法 | 0.0015232622 | 0.0914547032 |

表 2　训练后的样本数据

| $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ |
|---|---|---|---|---|
| 0.80403867(0.8) | 0.90457939(0.9) | 0.50335848(0.5) | 0.50366406(0.5) | 0.80917567(0.8) |
| 0.10000000(0.1) | 0.20000000(0.2) | 0.40000000(0.4) | 0.20000000(0.2) | 0.10000000(0.1) |
| 0.80774656(0.8) | 0.70825863(0.7) | 0.40719871(0.4) | 0.90751822(0.9) | 0.52527802(0.5) |
| 0.30526613(0.3) | 0.80604836(0.8) | 0.20426451(0.2) | 0.60470906(0.6) | 0.61138852(0.6) |
| 0.89658513(0.9) | 0.49606004(0.5) | 0.49725228(0.5) | 0.9969603428(1) | 0.49303938(0.5) |
| 0.90218034(0.9) | 0.60209515(0.6) | 0.60234743(0.6) | 0.80232454(0.8) | 0.40991389(0.4) |
| 0.88273533(0.9) | 0.78144562(0.8) | 0.88417934(0.9) | 0.88339308(0.9) | 0.9451973838(1) |
| 0.20266229(0.2) | 0.50283347(0.5) | 0.30248345(0.3) | 0.60259135(0.6) | 0.40870713(0.4) |
| 0.59563894(0.6) | 0.89512856(0.9) | 0.79626508(0.8) | 0.69597032(0.7) | 0.88833464(0.9) |
| 0.39984108(0.4) | 0.69954733(0.7) | 0.80026163(0.8) | 0.60011091(0.6) | 0.50312197(0.5) |
| …… | …… | …… | …… | …… |

## 5 小结

本文从一个全新的角度考虑 Petri 网的学习训练，提出了将结果反馈的 FPN 算法，在调整权值、变迁的阈值和变迁的可信度的同时调整输入矢量，多重作用来最小化误差函数，更有效地提高 FPN 的学习性能。实验结果表明该算法具有良好的学习效率和泛化能力。

### 参考文献


1 CHEN S. Weighted Fuzzy Reasoning Using Weighted Fuzzy Petri Nets. IEEE Transactions on Knowledge and Data Engineering (S1041-4347), 2002,14(2): 386－397.

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

3 陈亚华,吴时霖. 基于模糊有色 Petri 网的不确定性学习和推理方法. 系统仿真学报, 2003,15(增刊):56－58.

4 汤新民,钟诗胜. 基于元模型的模糊Petri网反向传播学习算法.系统仿真学报, 2007,14(19):3163－3167.

5 黄光球, 赵阿妮. 基于多值逻辑 Petri 网的攻击模型. 计算机工程, 2010,36(4):121－124.

6 Li Xiao ou, YU Wen, Lara-Rosano Felipe. Dynamic Knowledge Inference and Learning under Adaptive Fuzzy Petri Net Framework. IEEE Transactions on System, Man, and Cybernetics—Part C: Applications and Reviews (S1094-6977), 2000,30(4):442－450.

7 王万良. 人工智能及其应用. 北京: 高等教育出版社, 2008.

8 武妍,王守觉.一种通过反馈提高神经网络学习性能的新算法. 计算机研究与发展, 2004,9(41): 1488－1492.