

神经规则与案例融合的专家系统知识表示和推理 *

姚 路,康剑山,曾 斌
(海军工程大学,武汉 430033)

摘 要 :为了对专家系统中的知识进行表示,并使得推理变得更加准确高效,介绍了一种结合符号规则、神经网络和案例的方法。该方法是把一种混合规则,即神经规则和案例结合在一起。它与传统的基于符号规则推理不同的是,即使有一些未知的输入,也可以执行基于神经规则的推理。实验证明,与传统的专家系统比较,混合推理可以提高故障诊断的效率和精确度。

关键词 :神经规则,案例检索,混合推理,故障诊断

中图分类号 :TM344.1

文献标识码 :A

The Knowledge Representation and Reasoning of Expert System Used Neurule and Case Fusion

YAO Lu, KANG Jian-shan, ZENG Bin

(Naval University of Engineering, Wuhan 430033, China)

Abstract: To express the knowledge of expert system, and makes the reasoning become more accurate and efficient, this paper introduces an approach that integrates symbolic rules, neural networks and cases. The method is to mix a rule that neural rules and case together. The difference between this method and the traditional rules of reasoning based on symbol, even if there are some unknown input, it can also perform reasoning based on neural rules. Experiments show that, compared with the traditional expert system, hybrid reasoning can improve the efficiency and accuracy of fault diagnosis.

Key words: neurule, case indexing, hybrid reasoning, fault diagnosis

引 言

目前故障诊断方法主要有以下几种:^[1]基于故障树的方法、基于案例推理的方法、基于模型的方法、基于专家系统的方法、基于神经网络的方法等。以上方法各有其优缺点,为了提高推理系统的精度和效率,研究者往往希望整合两种以上的推理方法以提高混合形式的积极影响,同时减少它们的消极影响。在此文中介绍了一种有效的方式——神经规则,整合了 3 种推理计算方案,即基于规则推理,神经计算和案例推理,并将神经规则与案例结合起来。基于神经规则推理的应用领域大多在医学方面,而在装备故障诊断方面应用较少^[2]。

1 混合推理结构

基于神经规则和案例推理方法的混合系统构架如图 1 所示。该系统(虚线框部分)由以下几个模

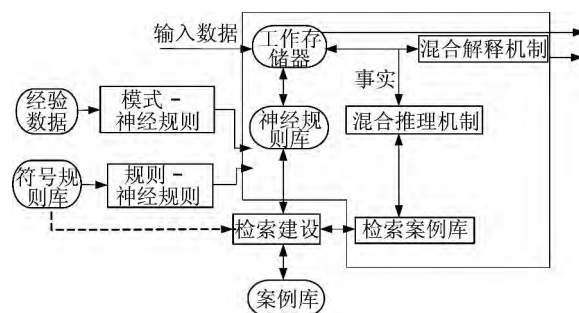


图 1 混合系统构架

收稿日期 2013-07-09

修回日期 2013-09-07

* 基金项目:国家自然科学基金(71201172);湖北省科技计划自然科学基金资助项目(ZRY0145)

作者简介:姚 路(1979-),男,湖北利川人,硕士,副教授。研究方向:装备管理、信息管理。

块组成:工作存储器,混合推理机制,混合解释机制,神经规则库和检索案例库。神经规则库含有代表一般领域知识的神经规则。这些神经规则既可以通过规则-神经规则模块从现存符号规则库产生。也可以通过模式-神经规则模块从现存的经验数据中产生。

使用神经规则库来检索某个案例库从而导出检索案例库。检索案例库中的这些案例是作为神经规则的异常案例。每个案例都需要将其形式化为一组属性-值对。对于某一案例,需要把它的输入属性和输出属性区分开来:输出属性代表观测的结果或做出的决定,并与其相关联的神经规则结论变量相匹配。其余属性是输入属性。案例一般可由以下形式的元组表示:

$$[i_1, i_2, \dots, i_m, O]$$

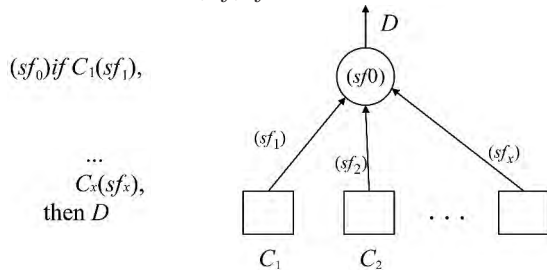
其中 $i_j (j=1, m)$ 表示输入属性的值, O 表示案例输出属性值。

2 神经规则概述

神经规则是一种混合规则。该方法利用神经计算在符号体系中以改善符号规则的性能^[3]。与其他的混合方法^[4]不同,利用这种方法构建的知识库保留了规则模块化,在很大程度上保留了自然性。而对于结论的解释^[5]通过产生式规则给出。

2.1 语义和语法

神经规则的形式见图 2(a)。其中 C_i 代表神经规则的前提, sf_i 是 C_i 的重要因子。 sf_0 是规则的偏差因子。神经规则也可以看作是自适应神经单元的形式(见图 2(b)), 其中 C_i, sf_i, sf_0 所表示的含义与前者相同。



(a) 神经规则形式 (b) 以自适应单元为形式的神经规则

图 2 神经规则形成和以自适应单元为形式的神经规则

在规则中输入值从 $[1, 0, 0.5]$ 选取值。其中, 1、0、0.5 分别代表前提的属性为真、假、未知 3 种情况。与符号规则相比, 这样取值更有利于区分前提的属性, 特别是当前提为假或未知时。符号 D 代表规则的输出结论。

$$D = f(a) \quad (1)$$

$$a = sf_0 + \sum_{i=1}^n sf_i C_i \quad (2)$$

$$f(a) = \begin{cases} 1, & a \geq 0 \\ -1, & a < 0 \end{cases} \quad (3)$$

式(1)中, a 是激活值, 阈值函数 $f(x)$ 是激活函数。输出值(“1”“-1”)分别代表规则成功或者失败。

2.2 神经规则库建设

神经规则可由生成的符号规则库中导出, 也可由经验数据导出。符号规则库中的符号规则的语法规则应与图 2(a)中相同, 但没有重要因子和偏差因子。

①从现存的符号规则库中导出神经规则^[3]

首先, 将含有相同结论的符号规则整合为合并集, 其次, 为每个合并集分配一个自适应神经单元, 每个单元通过著名的 LMS 算法进行训练。最终得到相应的神经规则。

例如, 在图 3 中列出了一份从波音公司的 CFM56-7 型发动机维修中提取的知识规则(发动机部位为 HPC), 其中符号规则 $R1, R2, R3$ 含结论相同, 三者可构成合并集, 图 4 给出了利用合并集产生的神经规则(NR1)。如果对符号规则和神经规则输入相同的初始数据, NR1 和符号规则 $R1 \sim R3$ 将产生相同的结论。因此, NR1 可以代替 $R1 \sim R3$ 。通过这种方式有效减小规则库的规模。

R1	R2	R3
if 部位明细=前缘, 损伤类型=裂纹, 尺寸大小>0.8 mm, 形状复杂度>6 then 损伤评估是一般超标	if 部位明细=后缘, 损伤类型=裂纹, 形状复杂度>6 then 损伤评估是一般超标	if 部位明细=后缘, 损伤类型=凹坑, 尺寸大小>0.8 mm, 形状复杂度>6 then 损伤评估是一般超标

图 3 符号规则合并集

NR1 (-7.8) if 部位明细=后缘(6.9), 损伤类型=裂纹(6.4), 形状复杂度>6(6.3), 部位明细=后缘(3.0), 尺寸大小>0.8 mm(2.7), 损伤类型=凹坑(2.6), then 损伤评估是一般超标

图 4 符号规则合并集从表 1 产生的神经规则

②从经验数据中导出神经规则^[4]

经验数据一般是以知识模式的形式存储。一个知识模式中含有大量变量和元组。每个变量值从一组离散值中选取。每个模式都有如下形式:

$$[v_1, v_2, \dots, v_n, d]$$

其中 d 代表中间变量或输出变量值, $v_i (i=1, n)$ 代表输入变量值。例如, 在针对某型舰船发动机的诊断知识进行处理, 并对变量值约简处理^[5]后得到 10 个知识模式(见下页图 5), 包含输出变量“故障类型”和输入变量(“冷却水温”、“空气流量”、“燃油压力”)。

由于知识模式可能产生非线性的情况, 所以对于某一输出变量值可能会产生多个神经规则。

例如, 从图 5 的知识模式中产生的神经规则如

冷却水温	空气流量	燃油压力	故障现象
1	1	2	正常
2	2	2	喷油器故障
1	3	1	正常
2	1	2	喷油器故障
3	2	3	点火故障
1	3	3	喷油器故障
3	1	2	点火故障
2	2	1	喷油器故障
2	3	3	点火故障
3	1	1	喷油器故障

图 5 变量“故障类型”知识模式

图 6 所示,从表中可以看出分别只有一个神经规则对应“故障类型”的“正常”值(NP4)和“点火故障”值(NP1),但是有两个神经规则对应“喷油器故障”值(NP2,NP3)。这是因为在“喷油器故障”值形成的相关训练组是非线性组。

NP1 (-0.5) if 燃油压力=1(-4.6), 冷却水温=1(-3.8), 冷却水温=3(3.0), 燃油压力=3(2.8), 空气流量=1(-2.0), 空气流量=2(0.9), 空气流量=3(-0.7), 冷却水温=2(-0.3), 燃油压力=2(-0.3), then 故障类型是点火故障	NP3 (-2.6) if 冷却水温=2(-6.2), 空气流量=2(-6.0), 燃油压力=2(-5.7), 燃油压力=3(4.7), 空气流量=1(3.2), 燃油压力=1(-2.7), 冷却水温=3(2.6), 冷却水温=1(-1.0), 空气流量=3(-0.2), then 故障类型是喷油器故障
NP2 (-1.0) if 燃油压力=3(-4.1), 冷却水温=3(-2.8), 空气流量=3(-2.2), 燃油压力=2(1.6), 空气流量=2(1.5), 冷却水温=2(1.3), 燃油压力=1(1.3), 空气流量=1(-0.6), 冷却水温=1(-0.4), then 故障类型是喷油器故障	NP4 (-0.8) if 冷却水温=1(3.2), 燃油压力=3(-2.7), 冷却水温=2(-2.6), 空气流量=2(-2.6), 冷却水温=3(-2.5), 空气流量=1(1.4), 燃油压力=2(1.0), 空气流量=3(-0.7), 燃油压力=1(-0.3), then 故障类型是正常

图 6 从表 3 中产生的神经规则

2.3 基于神经规则推理

下面介绍一种启发式算法,这种算法可以进一步提高基于神经规则推理的效率。首先,如果一个前提评估为“真”(例如图 6,在 NP2 中“燃油压力=3(-4.1)”),那么未评估的同名前提(含有相同变量)不计入 rs ,因为它们显然为“假”,所以它们的结果为 0。另外,未评估的同名前提中只有重要因子绝对值最大的前提才计入 rs 。例如,在 NP1(图 6)中,前提“燃油压力=3(2.8)”经过评估后,只有前提“空气流量=1(-2.0)”计入未评估总数。因为其同名前提“空气流量=2(0.90)”和“空气流量=3(-0.70)”重要因子绝对值没有前者大,而这 3 个前提不可能同时为真。同样剩下的两个前提也不计入未评估总数,因为它们已经确定结果为假(其同名前提“燃油压力=3(2.8)”已被评估为真)。

为了详细说明推理过程,假设现在有一个含有图 6 的 4 个神经规则的知识库和一个工作存储器,该存储器存储了如下事实 {“冷却水温=1”“空气流量=3”“燃油压力=3”}。推理引擎按照顺序开始检查每个神经规则以判断某个神经规则能否被激活。评估神经规则的前提是按照其存储顺序进行的。每个

前提评估后,将会对每一个神经规则的已评估总数和未评估总数进行计算。临界前提(括号内已给出)评估后每个神经规则的 ks 和 rs 值如下:

NP1: $ks_1=-1.5, rs_1=0.9$ (失败, 因为 $|ks_1|=1.5 > rs_1, ks_1 < 0$)

NP2: $ks_2=-7.3, rs_2=1.3$ (失败, 因为 $|ks_2|=7.3 > rs_2, ks_2 < 0$)

NP3: $ks_3=2.1, rs_3=1.2$ (成功, 因为 $|ks_3|=2.1 > rs_3, ks_3 > 0$)

NP3 成功,推理进程终止,输出结论是“故障类型是喷油器故障”,并把结论存放到工作存储器中。可以看出,得出结论并不需要评估所有的神经规则前提。

实验结果表明,基于神经规则推理比相应的基于符号规则推理更有效。主要原因在于神经规则是通过多个含有相同结论的符号规则形成的,这样有效减少了神经规则库中的规则数目和前提总数。这种推理机制使得推理过程中需要评估的规则与前提数目减少。启发式算法进一步强化了这种推理方法。

与基于符号规则推理相比,基于神经规则推理还有另外一个优点:即使一些前提未知(即局部输入),仍然可以执行基于神经规则推理并得到正确结论^[7]。而在基于符号规则推理中只有前提已知才能得出最终结论。

3 检索机制

为了执行基于神经规则和案例的混合推理方法,需要利用检索机制来组织合适的案例。在这里利用神经规则库中的神经规则来检索异常案例。

3.1 直接检索

直接检索的目的是为了将可用的神经规则和可用的异常案例关联起来,这些案例构成神经规则的异常。直接检索过程如下:

对每一个案例:

①将案例的输入属性值作为初始数据,执行基于神经规则推理。

②如果神经规则能够被激活,检查其结论值是否与对应的案例输出属性值相符合。如果不符合,那么将该案例视为神经规则的异常案例。

NR2 (-13.1) if 损伤类型=凹坑(6.9), 尺寸大小>1.6 mm(5.2), 尺寸大小>1.2 mm(4.8), 部位明细=后缘(2.7), 部位明细=前缘(1.6), then 损伤评估是严重超标

图 7 神经规则举例

下面举例说明如何进行检索过程,在这里使用图 7 给出的神经规则 NR2(使用与 NR1 类似方法产

生的)和图 8 中的两个案例。图 8 中只列出了案例最重要的属性。其他属性未列出。

序号	部位明细	损伤类型	尺寸大小	形状复杂度	重要程度	损伤评估
1	后缘	凹坑	>1.2 mm	<4	<4	一般超标
2	前缘	凹坑	>1.6 mm	>8	<4	严重超标

图 8 案例举例

设 ks_1 和 rs_1 分别表示第 1 个案例中神经规则的已评估总数和未评估总数。同样,设 ks_2 和 rs_2 分别表示第 2 个案例中神经规则已评估总数和未评估总数。只有当 $|ks_1| > |rs_1|$ ($|ks_2| > |rs_2|$) 第 1(2) 案例前提的评估才会终止。前文指出当一个前提评估为真,其值为 1,反之为假,值为 0。最终已评估总数和未评估总数的值是:

$$ks_1=1.3>0 \quad rs_1=1.6$$

$$ks_2=0.6>0 \quad rs_2=0$$

因为 ks_1 和 ks_2 值均大于零,所以两个案例基于神经规则推理的结果都应该是“损伤评估是严重超标”。但是事实上只有第 2 个案例的损伤评估变量输出值与神经规则的结论相一致,第 1 个案例对应的输出值与结论相矛盾。因此,该案例被检索为 NR2 的异常案例。

3.2 间接检索

只有对符号规则库关联的异常案例才能够使用间接检索^[8]。检索过程如下所示:

①通过规则-神经规则模块将符号规则转变为神经规则。

②将产生的神经规则与属于原符号规则库中关联的异常案例结合起来。

例如,对于图 3 中给出的 3 个符号规则(R_1, R_2, R_3),图 9 给出了与该规则库相关联的异常案例。表中“损伤评估”列的括号内标出了每个异常案例所对应的符号规则。此时图 9 中的案例被检索为 NR1 的异常案例。

序号	部位明细	损伤类型	尺寸大小	形状复杂度	重要程度	损伤评估
1	前缘	裂纹	>0.8 mm	>6	>4	严重超标(R1)
2	后缘	裂纹	<0.8 mm	>6	>4	严重超标(R2)
3	后缘	凹坑	<0.8 mm	>6	<2	正常 (R3)
4	后缘	凹坑	>0.8 mm	>6	>4	严重超标(R3)

图 9 图 5 中符号规则检索的异常案例

可以看出 NR1 中的异常案例要比在合并集中平均每个符号规则关联的异常案例数目多(NR1 关联 4 个异常案例,每个符号规则异常案例数目<2)。因此,平均来看,神经规则有一个明显的缺陷,即神经规则关联后的异常案例数目要多于产生该神经规则的符号规则库中每个符号规则关联的异常案例数目。这对基于案例推理的过程可能造成一定的影响。但是这种缺陷可以通过基于神经规则解

释机制来弥补。

4 诊断实例

为了测试方法的有效性,使用一个关于某型装备故障诊断的符号规则库。通过对专家访谈等方法,该符号库共收录了 58 种符号规则。这种结合了符号规则库和检索案例库的方法简称为 SRCL。符号规则可通过规则-神经规则模块产生神经规则。过程执行完毕后共产生了 34 个神经规则(即规则数目减少了 41%,前提数目减少了 30%)。上文指出,与符号规则库相关联的异常案例通过产生的神经规则进行间接检索。这种结合了神经规则库和检索案例库的方法简称为 NRCL。

推理序号(SRCL/NRCL)	访问规则(SRCL)	评估前提(NRCL)	异常出现
1	8/4	17/10	No
2	11/6	26/19	Yes
3	13/8	26/22	No
4	17/10	34/27	No
5	20/10	28/19	No
6	21/10	32/24	No
7	22/10	31/23	No
8	23/10	35/22	No
9	24/11	43/31	No
10	25/12	47/37	No
11	26/13	48/39	Yes
12	29/15	57/44	Yes
13	30/15	59/46	Yes
14	31/15	63/42	No
15	32/16	58/43	Yes
16	34/18	70/18	No
17	35/19	64/49	Yes
18	41/23	77/57	No
19	42/24	77/66	No
20	43/24	78/63	Yes
21	44/24	78/58	No
22	47/26	86/71	No
23	48/27	86/66	No
24	49/27	90/64	Yes
25	51/28	76/60	No
26	54/30	72/57	No
27	820/435	1458/1107	

图 10 实验结果

SRCL 和 NRCL 代表了两种推理方法。运用 SRCL 执行整合了基于规则推理和基于案例推理的混合推理机制。运用 NRCL 执行整合了基于神经规则推理和基于案例推理的混合推理机制。对于相同

(下转第 125 页)

的操作暂停,该盟员上的对象类属性暂不变化。CF 从记录文件中调出 T_i 时刻 proxy_tank 的属性值,并更新 proxy_tank。

(5) CF 上 proxy_tank 调用 AcquireOwnership(),TF 上 orig_n_tank 的 ReleaseOwnership() 会被触发,坦克属性所有权由 TF 转移到 CF。

(6) CF 上 proxy_tank 调用 update(),TF 上 orig_n_tank 收到属性更新通知,属性更新为 T_i 时刻状态。

(7) TF 上 orig_n_tank 调用 AcquireOwnership(),CF 上 proxy_tank 的 ReleaseOwnership() 会被触发,坦克属性所有权由 CF 转移到 TF。

(8) CF 发出恢复演习通知,TF 继续返回到正常演习状态,继续更新 orig_n_tank 属性。

5 结束语

本文从基于 HLA 的仿真训练系统解决数据记录与回放的问题出发,对多种可行方案进行了分析比较,提出了一种基于“事件”机制和 HLA 对象管理、所有权管理服务的技术方案,满足了模拟训练实时性和可靠性等要求,基于该方案的子系统在超过 120 个席位的仿真对抗中使用正常,收到了较好的效果。目前数据记录采用“集中”式记录方法,所有数

据通过 RTI 服务器传送至数据记录联邦成员,随着仿真规模的增大,可能会造成资源分布和负载不平衡,引起网络延迟等,需要今后进一步研究解决。

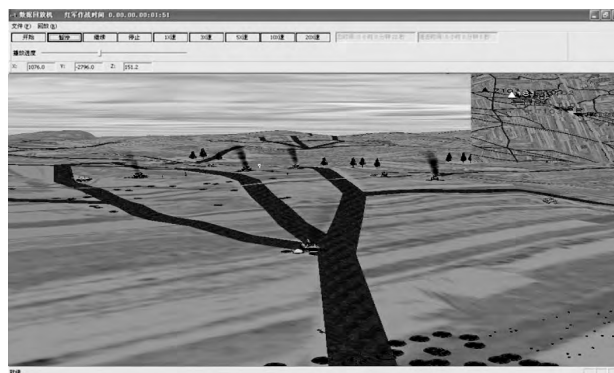


图 7 系统记录与回放效果图

参考文献:

- [1] 吴跃平. 基于 HLA 的数据记录与回放系统的研究与实现[D]. 成都:西南交通大学, 2012:41-62.
- [2] 张新宇. HLA 联邦回放及其通用性实现的一种方法[J]. 系统仿真学报, 2007, 19(22): 5188-5189.
- [3] 张家祥. HLA 仿真技术应用教程[M]. 北京:国防工业出版社, 2007:6.
- [4] 陈 彬. 一种基于 HLA 的通用仿真数据收集的方法[J]. 计算机仿真, 2006, 23(5): 127-129.

(上接第 120 页)

的初始数据运用 SCRL 和 NCRL 推理显然将产生同样的结论。然而运用 NCRL 推理与运用 SCRL 推理相比要求评估的前提数目更少。

图 10 给出了 SCRL 和 NCRL 推理的实验结果。给出的结果中含有访问规则的数目和已评估前提的数目。该表同样说明了结论是否由异常案例导出(“异常发生”列)。

从表中可以看出,在 NCRL 中访问过的规则平均下降 45%。而且运用 NCRL 推理评估前提数目减少了 25%。

5 结 语

本文介绍了故障诊断技术的方法及一种整合了符号规则和神经单元的混合规则,即神经规则,同时将神经规则和案例结合在了一起。整合了两种推理机制,介绍了完备的混合构架,检索机制等,形成了较为完备的混合推理机制。通过实验数据分析,证明这种方法有效提高了装备故障诊断的效率和精度。

参考文献:

- [1] 杨 军,冯振生,黄考利,等.装备故障诊断技术[M].北京:国防工业出版社,2004:14-19.
- [2] 关惠玲,韩 捷.设备故障诊断专家系统原理与实践[M].北京:机械工业出版社,2006:59-78.
- [3] Hatzilygeroudis I, Prentzas J. Neurules: Improving the Performance of Symbolic Rules [J]. International Journal on AI Tools, 2010, 9(1): 113-130.
- [4] Hatzilygeroudis I, Prentzas J. Constructing Modular Hybrid Knowledge Bases for Expert Systems[J]. International Journal on Artificial Intelligence Tools, 2011, 10(2): 87-105.
- [5] 杨 广,吴晓平.舰船装备故障诊断特征知识获取方法研究[J].船舶工程,2009,31(2):52-54.
- [6] Gonzalez A, Dankel D. The Engineering of Knowledge Based Systems Theory and Practice [J]. Englewood Cliffs, 2003, 15(1): 26-67.
- [7] Hatzilygeroudis I, Prentzas J. An Efficient Hybrid Rule-based Inference Engine With Explanation Capability[C]// Proceedings of the 14th International FLAIRS Conference, Menlo Park, CA: AAAI Press, 2001: 27-231.
- [8] Gallant S I. Neural Network Learning and Expert Systems [M]. Cambridge, MA: MIT Press, 2004.