

Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006 (25-28 сентября 2006г., Обнинск): Труды конференции. В 3-т. Т.3. -- М.: Физматлит, 2006. С. 1022-1030.

http://qai.narod.ru/Workshop/lipinsky_cai2006.pdf

УДК 004.94:519.254:004.023:004.8.032.26

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ МЕТОДОМ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ¹

Л.В.Липинский², Е.С. Семенкин³

В работе описываются алгоритмы генетического программирования, позволяющие автоматически генерировать структуры нейронных сетей и базы правил систем управления на нечеткой логике. Работоспособность подхода проверяется на тестовых и реальных задачах.

Введение

Интеллектуальные информационные технологии широко и успешно используются для решения различных практических задач. В принципе, можно уже говорить о массовом характере их внедрения. Однако проектирование информационных технологий, как правило, трудоемкий процесс, требующий больших временных затрат и материальных вложений. Каждый раз исследователь, решивший использовать для решения той или иной задачи технологии искусственных нейронных сетей (НС), сталкивается с вопросом об архитектуре нейронной сети. В большинстве случаев основной метод подбора структуры НС – метод проб и ошибок, который, разумеется, не может гарантировать оптимального решения. Проектирование базы правил (БП), являющейся основой экспертных систем (в том числе и систем на нечеткой логике), сложный творческий процесс, требующий больших затрат. Затраты эти связаны с тем, что время экспертов

¹ Работа выполнена при поддержке ФЦНТП по проектам № 2006-РИ-16.0/001 и 2006-РИ-19.0/001/377.

² 660014, г. Красноярск, пр. им. газеты «Красноярский рабочий», 31, СибГАУ, lipinskiyl@mail.ru

³ 660014, г. Красноярск, пр. им. газеты «Красноярский рабочий», 31, СибГАУ, saor_semenkin@sibsau.ru

высоко оплачивается, что существенно сужает круг разработчиков, которые могут себе это позволить. А взаимодействие инженера по знаниям и эксперта – длительный неформализованный процесс, цель которого извлечение знаний из эксперта, превращается во взаимное обучение. Все это мешает широкому использованию данных технологий на практике.

Применение генетического программирования (ГП) позволяет существенно упростить и ускорить процесс разработки интеллектуальных технологий за счет его автоматизации, а следовательно – высвободить интеллектуальные ресурсы экспертов и ученых для решения подлинно творческих задач.

Конструктивные подходы выбора топологии НС, описанные в литературе (Миркес 1999, Степанов 2000, Freeman 1991), сводятся в основном к двум алгоритмам. Первый из них заключается в том, что изначально исследователь задает сложную структуру НС, в которой нейронов и связей между нейронами с избытком хватает для решения конкретной задачи. После того, как НС настроилась на решение задачи, структура НС упрощается. Примером данного метода может служить метод контрастирования (Миркес 1999). Другой подход придерживается обратной идеи. Изначально сеть задается простой, и при необходимости (если сеть плохо решает задачу) добавляются нейроны и связи. Оба этих метода по сути дела являются локальным спуском в пространстве структур НС. В данной работе предлагается процедура выбора структуры НС, основанная на алгоритме генетического программирования.

Другим, не менее популярным, направлением интеллектуальных технологий являются экспертные системы, в том числе и экспертные системы на нечеткой логике. Важным (если не основным) этапом построения экспертных систем является извлечение знаний из эксперта (экспертов) и представление их в виде некоторой базы правил (БП). Существует множество методов извлечения знаний из эксперта (круглый стол, ролевые игры, мозговой штурм, интервью, диалог, экспертные игры, анализ документов, анализ литературы, и т.д., см. (Гаврилова и др., 2001)). На практике применяются все доступные методики, их комбинации и модификации. Процесс разработки экспертных систем занимает от трех лет и более. Большая часть времени тратится на процесс извлечения знаний. Кроме того, существует проблема поиска «хороших» экспертов и высокая стоимость их услуг.

В данной работе предлагается процедура автоматического проектирования БП, основанная на методе генетического программирования. Эксперт требуется на начальном этапе и на конечном. На начальном этапе – для формирования критерия, позволяющего в некотором количественном эквиваленте оценить насколько одна БП предпочтительнее другой, а на конечном – чтобы осмыслить полученную БП и усовершенствовать ее.

1. Выбор структуры НС методом генетического программирования

Для использования ГП необходимо решить две задачи – кодирование нейронной сети в виде дерева (в общем случае дерево может быть *n*-арным, но обычно выбирают бинарное) и выбор функции пригодности. Для кодирования НС при помощи дерева необходимо определить терминальное (Т) и функциональное (F) множества.

Выбранный способ кодирования должен позволять:

представлять сети с межслойными связями;

представлять сети с такими связями, когда не обязательно каждый нейрон предыдущего слоя связан с каждым нейроном последующего слоя;

на одном слое допускаются нейроны с различными активационными функциями.

Эти требования направлены на обеспечение более гибкого поиска структуры НС.

Предлагается следующий подход. Терминальное множество состоит из частей (блоков), из которых (по предположению исследователя) должна состоять нейронная сеть. Это могут быть нейроны различных видов (в том числе входные), нейроны, уже объединенные в некоторые блоки (являющиеся частью слоя), и т.п. Функциональное множество состоит из элементов, показывающих, каким образом соотносятся между собой элементы из Т. Поясним сказанное на примере. Пусть имеется следующий набор $T = \{slIn12, slIn34, sl2, sl3\}$, $F = \{+sl, >\}$. Здесь *sl12* – блок, состоящий из первого и второго входных нейронов, *sl34* – блок, состоящий из третьего и четвертого входных нейронов, *sl2* – блок, состоящий из двух нейронов с некоторыми активационными функциями *S1* и *S2*, *sl3* – блок, состоящий из трех нейронов с функциями активации *S3*, *S4* и *S5*. Тогда дерево, изображенное на рис. 1, соответствует нейронной сети, приведенной на рис. 2.

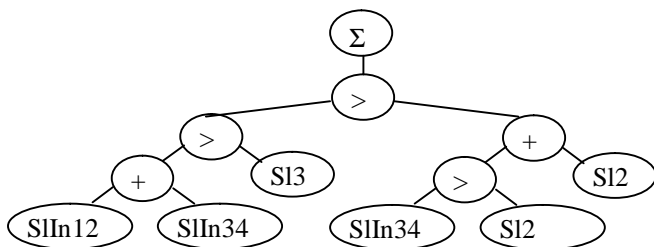


Рис. 1. Пример дерева, кодирующего НС

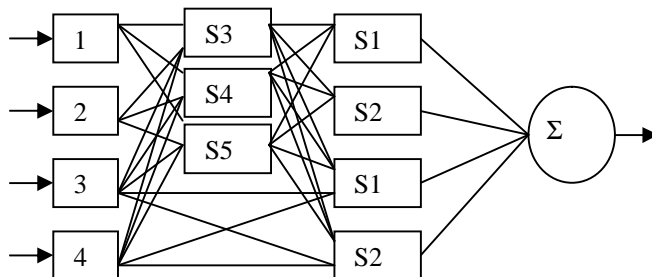


Рис. 2. НС, соответствующая бинарному дереву

Функции пригодности могут быть различными. Например, функция пригодности может быть равна ошибке НС после некоторого, наперед заданного, количества итераций настройки весовых коэффициентов.

2. Построение БП методом генетического программирования

БП ставит в соответствие набору значений входных параметров значение выходных параметров некоторой системы. Например, пусть существует некоторая система, описываемая двумя входными параметрами и одним выходным: $X=\{x_1, x_2, x_3\}$, $Y=\{y_1, y_2, y_3\}$ – вход, $U=\{u_1, u_2, u_3\}$ – выход. Тогда БП может иметь такой вид:

if ($x_1 \& y_1$) **then** u_1 **else if** ($x_1 \& y_2$) **then** u_1 **else if** ($x_1 \& y_3$) **then** u_1 **else if** ($x_2 \& y_1$) **then** u_2 **else if** ($x_2 \& y_2$) **then** u_1 **else if** ($x_2 \& y_3$) **then** u_3 **else if** ($x_3 \& y_1$) **then** u_2 **else if** ($x_3 \& y_2$) **then** u_1 **else if** ($x_3 \& y_3$) **then** u_3 .

Данная БП является «полной», т.е. перечислены все возможные комбинации входных параметров, и им в соответствие ставится значение выходного параметра. Такие БП, как правило, очень объемные и, по сути, представляют собой модель черного ящика. На практике чаще встречаются «упрощенные» БП. Например, следующая БП эквивалентна исходной, однако, содержит меньше правил, и более понятна человеку (эксперту):

if (x_1) **then** u_1 **else if** (y_2) **then** u_1 **else if** (y_1) **then** u_2 **else if** (y_3) **then** u_3 .

Применение ГП позволяет создавать «упрощенные» БП. Выбор терминального множества в данном случае определяется однозначно: значение выходных параметров и их комбинации. В нашем случае терминальное множество будет $T=\{u_1, u_2, u_3\}$. Выбор функционального множества, напротив, может решаться различными способами. Функциональный элемент делит по какому-либо правилу вектор входных параметров на два вектора (если для кодирования используются бинарные деревья) и более. Для кодирования указанной («упрощенной») БП достаточно следующего

функционального множества: $F=\{\%X, \%Y\}$. Оператор $\%X$ делит вектор значений параметра X пополам (если количество элементов четное) или с перевесом в один элемент (если количество элементов нечетное). Аналогично действует оператор $\%Y$. Тогда БП можно представить в следующем виде (рис. 3).

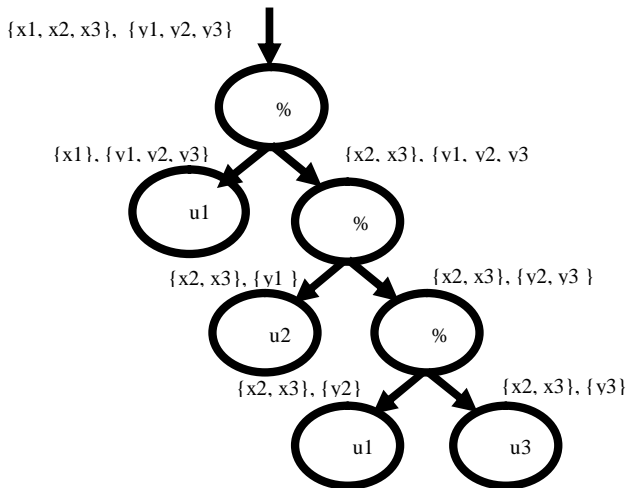


Рис. 3. Представление упрощенной БП в виде дерева

Решив вопросы представления информации, можно переходить непосредственно к процедуре ГП (Коца 1992).

3. Тестирование подхода

Прогнозирование технического состояния турбины по показаниям вибрационных датчиков. Задача имеет 11 входов 12 выходов. Необходимо построить нейросетевую модель, прогнозирующую состояние турбины. Из основной выборки (объемом 1000 элементов) случайным образом взяли 10% измерений и поместили в тестовую выборку. Нейронная сеть (НС) создавалась по оставшейся выборке (900 элементов) и проверялась на тестовой выборке. Ошибка НС вычислялась по формуле:

$$E = \frac{1}{N} \sqrt{\sum_{i=1}^N (x_i - x_i^*)^2} \quad (3.1)$$

Здесь, N – количество примеров в выборке, x_i – i -й прогноз нейронной сети, x_i^* – i -е реальное значение.

При тестировании были получены следующие результаты: на обучающей выборке минимальная ошибка - 0.001, максимальная – 0.035, средняя по всем выходам – 0.09; на тестовой выборке, соответственно 0.002, 0.084, 0.027.

Прогнозирование деградации электрических характеристик солнечных батарей. Необходимо по имеющимся результатам измерения параметров солнечных батарей (БС) в полёте космического аппарата (КА) (измерялись параметры секций БС3 и БС4 на КА ЭКСПРЕСС-А №2) и результатам измерения параметров БС ЭКСПРЕСС-А и GOES в годы активного Солнца (25 событий за период с 04.04.00 по 22.11.01, от экстремально мощных 14.07.00 и 22.11.01 до обычных - 16.10.00) построить нейросетевую модель, прогнозирующую деградацию электрических характеристик БС.

Нейросетевая модель настраивается на определение электрических характеристик солнечных батарей в зависимости от следующих факторов (входной вектор):

- интегральный флюенс протонов с различными энергиями (от 1 до 100 MeV);
- интегральный флюенс электронов с различными энергиями (от 0,6 до 2 MeV);
- ресурс - это параметр, который задан как количество дней с момента контакта отделения КА, характеризует повреждения от метеоритных тел и от ультрафиолетового излучения;
- коэффициент освещенности КА – величина, характеризующая степень освещенности аппарата, зависит от взаимного положения КА, Земли и Луны, имеет принципиальное значение, т.к. некоторые выходные характеристики, в частности сила тока короткого замыкания, зависят от освещенности БС.

Вышеперечисленным данным в соответствие ставятся: напряжение холостого хода U_{xx} и сила тока $I_{кз}$ БС (вектор значений или выходной).

Результат. Для обучения нейронной сети из выборки (объемом 220 записей) случайным образом было удалено 9% примеров (20 записей). На оставшейся выборке было получено четыре нейронных сети для прогнозирования $I_{БС3}$ (сила тока БС3), $I_{БС4}$ (сила тока БС4), $U_{хх_БС3}$ (напряжения холостого хода БС3), $U_{хх_БС4}$ (напряжения холостого хода БС4). Результаты тестирования приведены в таблице 1. Ошибка нейронной сети считалась по формуле (3.1).

Таблица 1.

Выход	Выборка	Ошибка
$I_{БС3}$	Обучающая	0.00210
$I_{БС3}$	Тестовая	0.00537

I_БС4	Обучающая	0.000935
I_БС4	Тестовая	0.00376
Uxx_БС3	Обучающая	0.000367
Uxx_БС3	Тестовая	0.00166
Uxx_БС4	Обучающая	0.000229
Uxx_БС4	Тестовая	0.00125

Автоматическое формирование БП. Оценка подхода производилась на тестовой задаче управления перевернутым маятником (Липинский и др. 2004). Система «тележка - перевернутый маятник» изображена на рис. 4. Состояние системы в каждый момент времени характеризуется значением ее параметров:

- положение системы – $x(t)$;
- линейная скорость системы – $x'(t)$;
- угол отклонения маятника – $\Theta(t)$;
- угловая скорость маятника – $\Theta'(t)$.

Система может перемещаться вдоль одной оси в интервале $[-100;100]$ м. Маятник может совершать колебания в интервале $[-90;90]^\circ$. Если значения положения системы или угла маятника превышают указанные интервалы, то считается, что система управления потерпела неудачу.

Необходимо построить нечеткий логический контроллер, управляющий системой. Целью управления является приведение системы в состояние равновесия, которое характеризуется нулевым значением отклонения маятника от вертикальной оси и нулевым значением позиции тележки, за счет передвижения системы вдоль оси X , при любом начальном допустимом положении тележки и маятника.

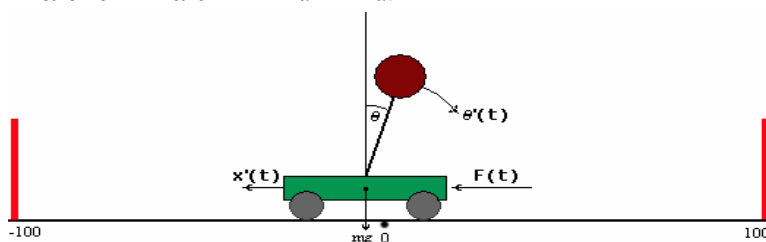


Рис. 4. Система «Тележка – Перевернутый маятник»

В результате работы программы (Липинский и др. 2006) была получена следующая база, состоящая из 8 правил:

IF (позиция отрицательная) AND (угловая скорость отрицательная большая) THEN сила положительная малая;

ELSE IF (позиция отрицательная) AND (угловая скорость отрицательная малая) THEN сила положительная большая;

ELSE IF (позиция неотрицательная) AND (угловая скорость отрицательная) THEN сила отрицательная большая;

ELSE IF (угол отрицательный) AND (угловая скорость неотрицательная) THEN сила нулевая

ELSE IF (скорость отрицательная) AND (угол положительный) AND (угловая скорость нулевая) THEN сила нулевая;

ELSE IF (скорость отрицательная) AND угол (положительный) AND (угловая скорость положительна) THEN сила положительная большая;

ELSE IF (позиция отрицательная) AND (скорость неотрицательная) AND (угол неотрицательный) AND (угловая скорость неотрицательная) THEN сила положительная большая;

ELSE IF (позиция положительная) AND (скорость неотрицательная) AND (угол неотрицательный) AND (угловая скорость неотрицательная) THEN сила положительная большая.

Здесь приняты следующие обозначения: позиция – позиция тележки; скорость – скорость тележки; угол – угол отклонения маятника; угловая скорость – угловая скорость маятника.

Рассмотрим данную БП. 1-е правило «говорит» о том, что если позиция отрицательная и угловая скорость отрицательная большая, то сила положительная малая. На первый взгляд данное воздействие помогает упасть маятнику. Однако, как только позиция тележки станет положительной, то в силу вступает правило 3, которое останавливает тележку рывком, поднимая при этом маятник в вертикальное положение.

Второе правило похоже на первое, только благодаря малой угловой скорости система управления прилагает большую положительную силу, что быстрее приводит систему к правилу 3.

Правила 4 и 5 «говорят» ничего не делать, если система находится в этих состояниях. В правиле 4 маятник либо движется в желаемое состояние, либо покоится, поэтому можно ничего не предпринимать. В правиле 5, казалось бы, необходимо приложить положительную силу, однако правило «призывает» не прикладывать силы. Дело в том, что в ситуации, когда у тележки скорость положительная, а угол отрицательный, вряд ли угловая скорость может быть равна нулю, по этому данное правило либо редко срабатывает, либо не срабатывает вовсе.

Правило 6 напоминает правило 5, только угловая скорость положительна, и в такой ситуации система управления принимает правильное решение – приложить силу большую положительную.

Правила 7 и 8 рывком заставляют маятник двигаться к нулевому положению.

Нельзя признать данную базу правил совершенной, но с задачей она справляется и содержит небольшое количество правил.

Выводы. Применение генетического программирования является эффективным средством автоматического проектирования интеллектуальных информационных технологий, позволяющим получить результаты, сопоставимые с результатами специалистов. Применение данного подхода позволит существенно сократить затраты на разработку интеллектуальных технологий, а значит – перейти к их массовому внедрению.

Список литературы

[Гаврилова и др., 2001] Гаврилова Т.А Базы знаний интеллектуальных систем. – СПб.: «Питер», 2001. - 384 с.

[Липинский и др. 2004] Липинский Л.В., Малько В.А. Подходы к формированию базы правил для нечетких систем управления // Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – Вып. 5. – 2004.

[Липинский и др. 2006] Липинский Л.В., Малько В.А., Семенкин Е.С. Интеллектуальные технологии автоматизации проектирования системы управления на нечеткой логике. – М.: ВНИИЦ, 2006. - № гос. рег. 50200601213.

[Миркес 1999] Миркес Е.М. Нейрокомпьютер, проект стандарта. - Новосибирск: "Наука", 1999.

[Степанов 2000] Степанов М.Ф., Степанова Т.В. Планирующие искусственные нейронные сети для решения задач теории автоматического управления // Труды четвертого международного симпозиума «Интеллектуальные системы». – Москва, 2000.

[Freeman 1991] Freeman J., Skapura D. Neural Networks. Algorithms, Applications, and Programming Techniques. – Houston: University of Houston at Clear Lake, Addison-Wesley Publishing Company, 1991.

[Koza 1992] Koza John R. «Genetic programming tutorial» URL: <http://www.genetic-programming.com/ganimatedtutorial.html>