ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.855.5

Т.В.Абрамова, научный сотрудник, Национальный исследовательский Томский государственный университет (e-mail: egs@sibmail.com)

ПРОЕКТИРОВАНИЕ НЕЙРО-НЕЧЕТКОГО ДЕРЕВА РЕШЕНИЙ

Для повышения точности классификации нечетких деревьев решений предлагается процедура адаптации параметров с помощью нейросетевого обучения. В прямом цикле нечеткие деревья решений строятся на основе алгоритма нечеткого ID3, в цикле обратной связи параметры нечетких деревьев решений адаптируются на основе стохастического градиентного алгоритма путем обхода обратно с листьев на корневые узлы. С помощью этой стратегии иерархическая структура нечетких деревьев решений остается фиксированной.

Ключевые слова: *классификация, нечеткие деревья решений, адаптация, нейросетевое обучение.*

***

В отечественной и зарубежной литературе описано применение деревьев решений как мощной эволюционной методологии при решении задач классификации и регрессии [1–5]. Как инструмент DATA MINING (обнаружение скрытых знаний из данных) они используются для поиска и извлечения понятных человеку интерпретируемых правил классификации. Отметим, что, в состав многих пакетов, предназначенных для интеллектуального анализа данных, уже включены методы построения деревьев решений, они являются прекрасным инструментом в системах поддержки принятия решений.

Дерево принятия решений – это дерево, в листьях которого стоят значения целевой функции, а в остальных узлах – условия перехода (к примеру “ПОЛ есть МУЖСКОЙ”), определяющие по какому из ребер идти. Если для данного наблюдения условие – истина, то осуществляется переход по левому ребру, если же ложь – по правому. Обычно каждый узел включает проверку одной независимой переменной. Иногда в узле дерева две независимые переменные сравниваются друг с другом или определяется некоторая функция от одной или нескольких переменных.

Если переменная, которая проверяется в узле, принимает категориальные значения, то каждому возможному значению соответствует ветвь, выходящая из узла дерева. Если значением переменной является число, то проверяется больше или меньше это значение некоторой константы. Иногда область числовых значений разбивают на интервалы (проверка попадания значения в один из интервалов).

Листья деревьев соответствуют значениям зависимой переменной, т.е. классам.

На рис.1 показано дерево классификации ирисов. Классификация идет на три класса (на изображении помечены - красным, синим и зеленым), и проходит по параметрам: длина\толщина чашелистика (SepalLen, SepalWid) и длина\толщина лепестка (PetalLen, PetalWid). Как видим, в каждом узле стоит его принадлежность к классу (в зависимости от того, каких элементов больше попало в этот узел), количество попавших туда наблюдений N, а также количество каждого класса. Так же не в листовых вершинах есть условие перехода - в одну из дочерних. Соответственно, по этим условиям и разбивается выборка. В результате, это дерево почти идеально (6 из 150 неправильно) классифицировало исходные данные (именно исходные - те на которых оно обучалось).

Ниже перечислены несколько основных методов, которые используют деревья принятия решений, их краткое описание, плюсы и минусы (табл.1).

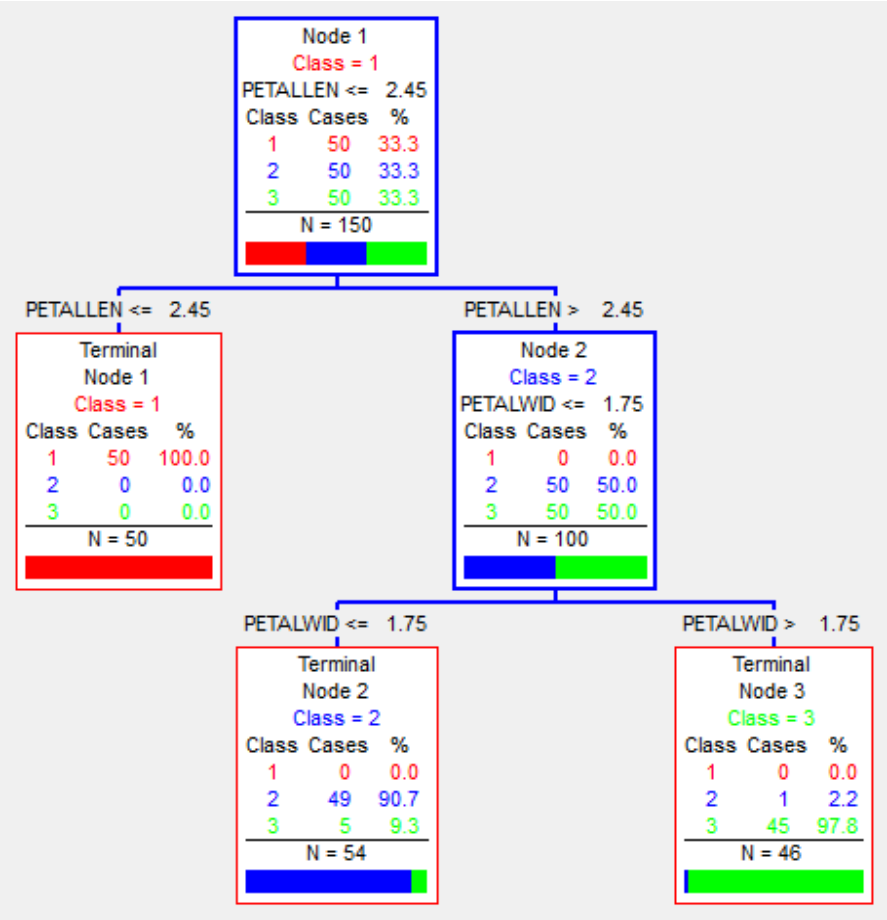

Рис.1. Дерево классификации ирисов

Таблица 1

Сравнительный анализ методов, использующих деревья решений

Метод	*Преимущества*	*Недостатки*
CART	Быстрое построение модели. Легко интерпретируется (из-за простоты модели, можно легко отобразить дерево и проследить за всеми узлами дерева)	Часто сходится на локальном решении (к примеру, на первом шаге была выбрана гиперплоскость, которая максимально делит пространство на этом шаге, но при этом это не приведёт к оптимальному решению)
Random forest	Высокое качество результата, особенно для данных с большим количеством переменных и малым количеством наблюдений. Возможность распараллелить. Не требуется тестовая выборка	Каждое из деревьев огромное, в результате модель получается огромная. Долгое построение модели, для достижения хороших результатов. Сложная интерпретация модели (Сотни или тысячи больших деревьев сложны для интерпретации)
Stochastic Gradient Boosting	Высокое качество результата, особенно для данных с большим количеством наблюдений и малым количеством переменных. Сравнительно (с предыдущим методом) малый размер модели, так как каждое дерево ограничено заданными размерами.	Требуется тестовая выборка (либо кросс-валидация). Невозможность хорошо распараллелить. Относительно слабая устойчивость к ошибочным данным и переобучению. Сложная интерпретация модели (Так же, как и в Random forest)

CART (англ. Classification and regression trees - Классификационные и регрессионные деревья) был первым из методов, придуманный в 1983г. четверкой известных ученых в области анализа данных: Leo Breiman, Jerome Friedman, Richard Olshen and Stone [2].

Суть этого алгоритма состоит в обычном построении дерева принятия решений. На первой итерации мы строим все возможные (в дискретном смысле) гиперплоскости, которые разбивали бы наше пространство на два. Для каждого такого разбиения пространства считается количество наблюдений в каждом из подпространств разных классов. В результате выбирается такое разбиение, которое максимально выделило в одном из подпространств наблюдения одного из классов. Соответственно, это разбиение будет нашим корнем дерева принятия решений, а листами на данной итерации будет два разбиения.

На следующих итерациях мы берем один худший (в смысле отношения количества наблюдений разных классов) лист и проводим ту же операцию по разбиению его. В результате этот лист становится узлом с каким-то разбиением, и двумя листами.

Продолжаем так делать, пока не достигнем ограничения по количеству узлов, либо от одной итерации к другой перестанет улучшаться общая ошибка (количество неправильно классифицированных наблюдений всем деревом). Однако полученное дерево будет “переобучено” (будет подогнано под обучающую выборку) и, соответственно, не будет давать нормальные результаты на других данных. Для того, чтобы избежать “переобучения”, используют тестовые выборки (либо кросс-валидацию) и, соответственно, проводится обратный анализ (так называемый pruning), когда дерево уменьшают в зависимости от результата на тестовой выборке.

Относительно простой алгоритм, в результате которого получается одно дерево принятия решений. За счет этого, он удобен для первичного анализа данных, к примеру, чтобы проверить на наличие связей между переменными.

Random forest (Случайный лес) - метод, придуманный после CART одним из четверки - Leo Breiman в соавторстве с Adele Cutler [3], в основе которого лежит использование комитета (ансамбля) деревьев принятия решений.

Суть алгоритма, что на каждой итерации делается случайная выборка переменных, после чего, на этой новой выборке запускают построение дерева принятия решений. При этом производится “bagging” - выборка случайных двух третей наблюдений для обучения, а оставшаяся треть используется для оценки результата. Такую операцию проделывают сотни или тысячи раз. Результирующая модель будет результатом “голосования” набора полученных при моделировании деревьев.

Stochastic Gradient Boosting (Стохастическое градиентное добавление) - метод анализа данных, представленный Jerome Friedman [4] в 1999 году, и представляющий собой решение задачи регрессии (к которой можно свести классификацию) методом построения комитета (ансамбля) “слабых” предсказывающих деревьев принятия решений.

На первой итерации строится ограниченное по количеству узлов дерево принятия решений. После чего считается разность между тем, что предсказало полученное дерево, умноженное на learnrate (коэффициент “слабости” каждого дерева), и искомой переменной на этом шаге. И уже по этой разнице строится следующая итерация. Так продолжается, пока результат не перестанет улучшаться. Т.е. на каждом шаге мы пытаемся исправить ошибки предыдущего дерева. Однако здесь лучше использовать проверочные данные (не участвовавшие в моделировании), так как на обучающих данных возможно переобучение.

Общим недостатком построения традиционных деревьев решений является требование определенности входных данных, которая достигается путем применения средневзвешенных значений входных параметров анализируемой технологии, что может привести к получению значительно смещенных точечных оценок показателей эффективности проектов. Также очевидно, что требование детерминированности входных данных является неоправданным упрощением реальности, так как любая технология характеризуется множеством факторов неопределенности: неопределенность исходных данных, неопределенность внешней среды, неопределенность, связанная с характером, вариантами и моделью реализации проекта, неопределенность требований, предъявляемых к эффективности технологии. Именно факторы неопределенности определяют риск технологии, то есть опасность потери ресурсов, недополучения доходов или появления дополнительных расходов.

Для повышения точности классификации автор предлагает использовать нейронечеткие деревья решений, обладающие свойством адаптации параметров с помощью нейросетевого обучения. В прямом цикле нечеткие деревья решений строятся на основе алгоритма нечеткого ID3 [5]. В цикле обратной связи параметры нечетких деревьев решений адаптируются на основе стохастического градиентного алгоритма путем обхода обратно с листьев на корневые узлы.

В качестве исходных данных будем использовать так называемые треугольные нечеткие числа с функцией принадлежности следующего вида (рис.2).

Эти числа моделируют высказывание следующего вида: “параметр **A** приблизительно равен α и однозначно находится в диапазоне $[a_{min}, a_{max}]$”.

В общем случае под нечетким числом понимается нечеткое подмножество универсального множества действительных чисел, имеющее нормальную и выпуклую функцию принадлежности.

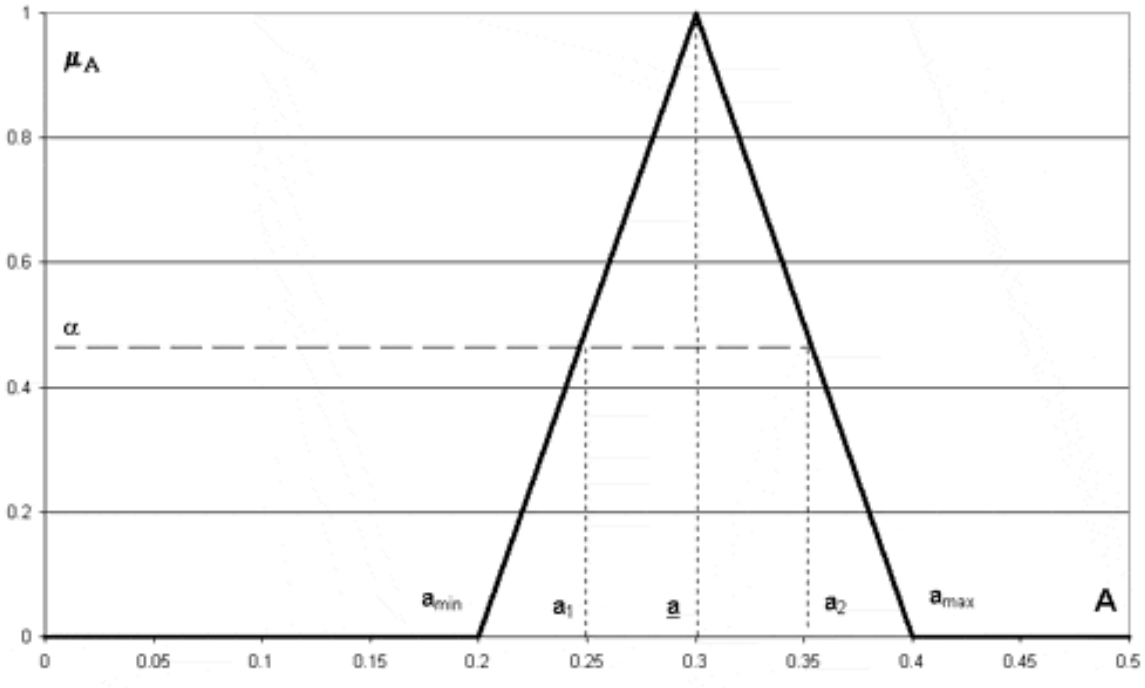

Рис. 2. Функция принадлежности треугольного нечеткого числа A

Такое описание позволяет экспертам взять в качестве исходной информации интервал параметра [a_{min}, a_{max}] и наиболее ожидаемое значение α, и тогда соответствующее треугольное число A = (a_{min}, α, a_{max}) построено. Выделение трех значимых точек исходных данных весьма распространено в инвестиционном анализе. Часто этим точкам сопоставляются субъективные вероятности реализации соответствующих (“пессимистического”, “нормального” и “оптимистического”) сценариев исходных данных. Далее будем называть параметры (a_{min}, α, a_{max}) значимыми точками треугольного нечеткого числа A.

Отметим, что атрибуты технологических инновационных проектов классифицируются как субъективные или объективные. Субъективные включают качественные характеристики такие, как технический уровень, преимущества предприятия, инновационный риск, управление проектом – будем оценивать их языковыми значениями, представленными нечеткими числами на основе экспертных опросов.

Объективные (количественные) признаки включают в себя проект инвестиционных затрат и т.д. Эти количественные признаки приводятся к общей шкале для обеспечения совместимости с языковыми значениями субъективных признаков. Рассмотрим для наглядности типовые описания атрибутов технологических проектов (табл. 2).

Первый атрибут (инвестиционные затраты по проекту) является численным. Третий и пятый атрибуты (преимущества предприятия и уровень управления проектом) описываются в лингвистических терминах, таких, как хорошо, очень плохо и т.д. Значение второго атрибута определено на нечетком множестве {1, 2, 3}. Инновационный риск на основе нечеткой классификации дается в лингвистических терминах.

Фазификация подразумевает перевод численных значений атрибутов в лингвистические термины в целях сокращения информации и представления ее в понятной для человека форме с целью принятия решений. Одним из способов определения функций принадлежности этих языковых переменных является мнение эксперта или восприятие людей.

Таблица 2

Атрибуты технологических проектов

№ п/п	Инвестиционные затраты по проекту (*10 млн руб.), x1	Технический уровень {1,2,3}, x2	Преимущества предприятия, x3	Инновационный риск, x4	Уровень управления проектом, x5
1.	0.82	0.0, 0.1, 0.3	среднее	низкий	Низкий
2.	0.81	0.0, 0.1, 0.9	среднее	средний	Средний
3.	0.78	0.0, 0.3, 0.5	хорошее	высокий	низкий
4.	1.00	1.0, 0.9, 0.0	довольно плохое	очень низкий	очень высокий
5.	0.97	1.0, 0.8, 0.6	плохое	средний	Средний
6.	0.80	0.0, 0.2, 0.9	очень плохое	низкий	Средний
7.	0.96	0.0, 0.4, 0.9	очень плохое	средний	очень высокий
8.	0.78	0.0, 0.1, 0.2	очень хорошее	средний	очень низкий
9.	0.98	0.0, 0.3, 0.8	довольно хорошее	средний	Средний
10.	0.78	1.0, 0.7, 0.4	хорошее	средний	довольно средний
11.	0.98	0.0, 0.2, 0.5	плохое	низкий	очень низкий
12.	0.81	0.0, 0.3, 0.9	хорошее	средний	Высокий

В целях автоматизации данной процедуры можно использовать статистические методы, а также нечеткую кластеризацию на основе самоорганизующегося нейросетевого обучения. Рассмотрим второй способ.

Пусть дан набор данных X, которые должны быть переведены в k лингвистических переменных T_j, j=1,2,…,k. Для простоты предположим, что функция T_j имеет вид триангуляции:

$$T_1(x)=\begin{cases}1,\ x\le a_1\\ (a_2-x)/(a_2-a_1),\ a_1<x<a_2\\ 0, x\ge a_2\end{cases}$$

$$T_j(x)=\begin{cases}0, x\ge a_{j+1}\\ (a_{j+1}-x)/(a_{j+1}-a_j),\ a_j\le x<a_{j+1}\\ (x-a_j)/(a_j-a_{j-1}),\ a_{j-1}<x<a_j\\ 0,\ x\le a_{j-1}\end{cases}$$

$$T_k(x)=\begin{cases}1,\ x\ge a_k\\ (x-a_{k-1})/(a_k-a_{k-1}),\ a_{k-1}<x<a_k\\ 0,\ x\le a_{k-1}.\end{cases}$$

Параметры, подлежащие определению по каждому атрибуту, образуют k центров $\{a_1, a_2,..., a_k\}$. Эффективным методом для определения этих центров является нейросетевой алгоритм – самоорганизующиеся карты Кохонена [3].

Рассмотрим численный атрибут проекта – инвестиционные расходы по группе примеров из таблицы 2. По самоорганизующимся картам Кохонена определим для него: a_1= 0.68, a_3= 0.76, a_3= 0.82.

Тогда функции принадлежности переменной *x* одной из лингвистических переменных T_j, (j=1,2,3) описываются следующим образом:

$$T_1(x)=\begin{cases}1,\ x\le 0.68\\ (0.76-x)/0.08,\ 0.68<x<0.76\\ 0,\ x\ge 0.76.\end{cases}$$

$$T_2(x)=\begin{cases}0,\ x\ge 0.82\\ (0.82-x)/0.06,\ 0.76<x<0.82\\ (x-0.68)/0.08,\ 0.68<x<0.76\\ 0,\ x\le 0.68.\end{cases}$$

$$T_3(x)=\begin{cases}1,\ x\ge 0.82\\ (x-0.76)/0.06,\ 0.76<x<0.82\\ 0,\ x\le 0.76.\end{cases}$$

Очевидно, эти лингвистические термины могут быть описаны как «низкие», «средние» и «высокие». Второй столбец таблицы 3 показывает степень близости атрибута «инвестиционные затраты» этим трем функциям принадлежности.

Для описания лингвистических и соответствующих им численных значений предполагаем, что функции принадлежности данного лингвистического термина известны.

Мера сходства между лингвистическими терминами может быть определена их функциями следующим образом:

f(A,B) = 0.5*{S(A,B) + S(B,A)}, где S(A,B) и S(B,A) представляют степень подмножественности A в B и B в A соответственно (подмножественность как степень принадлежности одного множества другому). Здесь подмножественность A в B определяется как $S(A,B) = M(A\cap B)/M(A)$, где M обозначает сумму степеней принадлежности перевода нечеткого множества в конечное состояние.

При помощи описанной функции f можно вычислить степень принадлежности каждой из двух лингвистических терминов «преимущества предприятия».

Значение нечетких атрибутов, например атрибута «технический уровень», может быть представлено функционально набором функций принадлежности (табл. 3). Для данного набора функций мы находим несколько новых нечетких множеств, которые рассматриваются как результат кластеризации исходных данных для описания множества функций принадлежности.

Таким образом, разработанная методика построения нейро-нечетких деревьев решений позволяет избавиться от средневзвешенных оценок входных данных и обладает свойством нейросетевой адаптации параметров на основе стохастического градиентного алгоритма путем обхода обратно с листьев на корневые узлы.

Таблица 3

Нечеткие множества данных технологических проектов после нейросетевой тренировки

N п/п	Инвестиционные затраты по проекту (*10 млн руб.),			Технический уровень {1,2,3}, x2			Преимущества предприятия, x3			Инновационный риск, x4			Уровень управления проектом, x5		
	низк	сред	выс	низк	сред	выс	хор	сред	плох	хор	сред	плох	высок	ред	низк
1	0.12	0.86	0.02	0.46	1.00	0.46	0.28	0.46	0.96	0.68	1.00	0.28	1.00	0.58	0.36
2	0.00	0.92	0.08	0.56	1.00	0.56	0.38	0.97	0.42	0.36	0.56	0.92	0.54	1.00	0.58
3	0.96	0.00	0.02	0.35	0.68	1.00	0.36	0.63	0.92	0.96	0.26	0.38	1.00	0.57	0.38
4	0.00	0.00	1.00	0.92	0.78	0.36	0.85	0.13	0.08	0.88	1.00	0.32	0.18	0.22	0.76
5	0.11	0.06	0.83	1.00	0.58	0.35	0.94	0.58	0.36	0.28	0.36	1.00	0.56	1.00	0.58
6	0.12	0.31	0.57	0.88	0.42	0.24	0.36	0.95	0.48	0.98	0.38	0.27	0.56	0.38	0.56
7	0.00	0.00	1.00	0.38	0.64	0.98	0.45	0.95	0.48	0.92	0.46	0.56	0.88	1.00	0.28
8	1.00	0.00	0.00	1.00	0.57	0.32	0.26	0.42	0.98	0.26	1.00	0.38	0.57	0.38	0.59
9	0.00	0.00	1.00	0.34	0.55	1.00	0.96	0.39	0.28	0.18	0.68	0.98	0.82	0.68	0.24
10	0.08	0.86	0.06	0.56	1.00	0.56	0.44	0.29	0.95	1.00	0.45	0.56	0.26	1.00	0.56
11	0.00	0.00	1.00	0.32	0.98	0.28	0.68	0.45	0.68	0.12	0.56	0.32	0.43	0.68	0.46
12	0.46	0.54	0.00	0.96	0.56	0.25	0.98	0.27	0.36	0.78	0.56	0.68	1.00	0.36	0.68

В прямом цикле нечеткие деревья решений строятся на основе алгоритма нечеткого ID3. В цикле обратной связи параметры нечетких деревьев решений адаптируются. С помощью этой стратегии иерархическая структура нечетких деревьев решений остается фиксированной.

В заключение отметим, что предлагаемый подход применения алгоритма обратного распространения непосредственно на структуре нечетких деревьев решений улучшает точность их обучения без ущерба для интерпретируемости.

Работа выполнена по программе повышения конкурентоспособности национального исследовательского Томского государственного университета.

Список литературы

1. Горбачев С.В., Сырямкин В.И., Койнов С.А. Интеллектуальная система стратегического бизнес-планирования с нечетко-множественной оценкой эффективности и рисков. – LAMBERT Academic Publishing, Saarbrucken, 2012. – 172 с.

2. Горбачев С.В., Сырямкин В.И., Рудаков И.Б. Распознавание сложнопостроенных залежей нефти, газа на основе нейро-нечетких портретов. – LAMBERT Academic Publishing, Saarbrucken, 2013. – 173 с.

3. Горбачев С.В., Сырямкин В.И. Нейро-нечеткие методы в интеллектуальных системах обработки и анализа многомерной информации. – Томск: Изд-во Томского государственного университета, 2014. – 510 с.

4. Горбачев С.В., Сырямкин В.И., Сырямкин М.В. Интеллектуальный Форсайт-прогноз научно-технологического развития государства. – LAMBERT Academic Publishing, Saarbrucken, 2012. – 132 с.

5. Janikow. Fuzzy Processing in Decision Trees // Proceedings of the Sixth International Symposium on AI. – 1993. P. 360-367.

Получено 01.12.15

T.V.Abramova, Senior Lecturer, Magnitogorsk State Technical University (e-mail: tanusha-atv@mail.ru)

DESIGN OF NEURO-FUZZY DECISION TREE

To improve the classification accuracy of fuzzy decision trees is proposed, the procedure of adapting parameters using neural network learning. In direct cycle, the fuzzy decision trees are built based on the algorithm of fuzzy ID3 tags, in the loop feedback parameters of fuzzy decision trees are adapted based on the stochastic gradient algorithm by traversing back from leaf to root nodes. Using this strategy, a hierarchical structure of fuzzy decision trees remains fixed.

Key words: *classification, fuzzy decision trees, adaptation, neural network learning.*