

## МЕТОД ГЕНЕРАЦИИ ТЕСТОВ ДЛЯ ОТЛАДКИ НЕЙРОСЕТЕВЫХ ЭКСПЕРТНЫХ СИСТЕМ

О.Н. Долинина, А.К. Кузьмин

*Кафедра «Прикладные информационные технологии»,  
ГОУ ВПО «Саратовский государственный технический университет»;  
olga@sstu.ru*

*Представлена членом редколлегии профессором В.И. Коноваловым*

**Ключевые слова и фразы:** искусственные нейросети; метод технической диагностики; отладка нейросетей; ошибки в базах знаний.

**Аннотация:** Приведены основные этапы жизненного цикла нейросетевых систем, анализ ошибок, возникающих в базах знаний экспертных систем; проанализированы существующие методы их отладки. Описан подход к решению задачи генерации исчерпывающих тестовых наборов, основанный на преобразовании структуры представления знаний экспертной системы к виду логической сети и применения к ней методов технической диагностики. Предложен метод тестирования нейросетевых экспертных систем, основанный на распространении данного подхода на модель многослойного персептрона.

---

### Введение

Из всего множества существующих экспертных систем можно выделить системы, в основе которых лежит нейросетевая модель представления знаний и механизм принятия решений [1]. Системы подобного класса обладают некоторыми особенностями. В частности, знания такой системы представлены не в явном виде (например, в виде продукционных правил), а в виде весовых коэффициентов связей между нейронами [2–4]. При этом значения коэффициентов настраиваются автоматически в процессе обучения, то есть в процессе их итеративной коррекции в сторону снижения суммарной ошибки на выходе системы. Для успешного обучения необходимо наличие обучающего множества, под которым понимают известную совокупность пар  $\langle X, Y \rangle$  входных и соответствующих выходных векторов системы. При этом роль экспертов сводится к созданию обучающего множества, а не к формализации своих знаний (последнее представляет собой сложную задачу, которая требует объединения значительных усилий когнитолога и эксперта предметной области). Создание обучающего множества заключается в определении списка входных признаков, которые оказывают влияние на результат, а также подборе пар входных и соответствующих выходных векторов. Далее полученные пары подаются на вход алгоритма обучения, который итеративно выполняет необходимые коррекции весовых коэффициентов.

## Решение задачи генерации тестовых наборов

Рассмотрим трехслойный персептрон  $P$ , обученный на множестве  $L$ . Входной вектор  $X$  соответствует множеству  $In$  значений входных нейронов. Персептрон  $P$  осуществляет отображение множества входных векторов  $X$  в множество выходных векторов  $Y: P = X \rightarrow Y$ .

Так как обучающее множество формируется экспертами, то не существует гарантии того, что обученная нейросеть не содержит ошибок. Введем понятие *ошибки нейросети*. Ошибка нейросети имеет место при наличии такой комбинации значений входных нейронов  $\{In_1, In_2, \dots, In_m \in In\}$ , при установке которой отображение  $P$  становится неверным.

Наиболее трудно обнаруживаемой является ошибка типа **«забывание об исключении»** [5, 6]. Определим данный класс ошибки в терминах нейросети. Формальная модель ошибки типа «забывание об исключении» заключается в том, что отображение  $X \rightarrow Y$  корректно всегда, кроме случая установки такой комбинации значений входных нейронов  $\{In_1, In_2, \dots, In_m \in In\}$ , при которой отображение  $P$  становится неверным.

Формальная модель ошибки типа **«забывание о нескольких исключениях»** заключается в том, что отображение  $X \rightarrow Y$  корректно всегда, кроме случаев установки таких комбинаций значений входных нейронов  $In = \{In_i, In_j, In_k, \dots\}$ , при которых отображение  $P$  становится неверным.

Ошибка типа «забывание об исключении» является наиболее общей и покрывает ошибку типа «забывание о нескольких исключениях». Единственным методом, позволяющим выявить все возможные ошибки в нейросети, является тестирование. Однако отсутствие хорошо разработанных методов и средств контроля качества нейросетевых систем сильно ограничивает возможности их применения [2]. Классическим подходом к тестированию нейросетей является разделение имеющейся базы примеров на обучающее, контрольное и тестовое множества. Обучающие примеры подаются на вход алгоритма обучения, при этом контрольное множество используется для контроля переобучения нейронной сети [2]. Тестовое множество в процессе обучения не используется и служит для тестирования уже обученной сети. Основная проблема заключается в том, что тестовое множество генерируется экспертами, что не позволяет обнаружить максимальное количество ошибок. Исходя из вышесказанного, необходимо разработать метод, позволяющий сформировать тестовое множество таким образом, чтобы тестирование гарантированно выявляло бы максимальное количество ошибок.

Для решения задачи генерации тестовых наборов предлагается подход, основанный на преобразовании нейронной сети к логической сети (ЛС) и применении к последней методов технической диагностики [9, 10]. В работе [5] представлена методология тестирования продукционных баз знаний при помощи алгоритма поиска константной неисправности в логической сети Path Oriented Decision Making (PODEM) [9]. В технической диагностике под константной неисправностью понимаются ошибки типа «константный ноль (единица)», когда на одном или нескольких выходах системы постоянно присутствует нулевой (единичный) сигнал независимо от входных значений [10]. Ошибку типа «забывание об исключении» можно рассматривать как неисправность «константный 0» логической сети, которая проявляется только при каком-либо наборе значений сигналов ЛС.

Также в работе [5] доказано, что если тест выявляет ошибки такого типа, то этот тест гарантированно выявляет ошибки и всех других типов.

Для генерации тестового набора нейросеть должна быть преобразована к продукционной базе знаний (ПБЗ), затем к виду И/ИЛИ-графа и затем к логической сети с целью дальнейшего использования методов технической диагностики цифровых устройств [5].

Продукционная база знаний определяется совокупностью

$$P = (F, R, G, C, I), \quad (1)$$

где  $F$  – конечное множество фактов о решаемой проблеме. Каждый факт может быть установленным или не установленным, совокупность установленных фактов задает некоторую ситуацию в предметной области. Множество продукций  $R$  или правил, включающее правила вида

$$r_m : \text{ЕСЛИ } f_i \text{ И } f_j \dots \text{ И } f_n \text{ ТО } f_k, \quad (2)$$

где  $r_m$  – имя правила,  $r_m \in R; f_i, f_j, \dots, f_n$  – условия выполнения правила;  $f_k$  – следствие правила,  $f_i, f_j, \dots, f_n, f_k \in F$ ;  $G$  – множество целей или терминальных фактов экспертной системы (ЭС);  $C$  – множество разрешенных комбинаций фактов, в которое не входят комбинации с установленными одновременно какими-либо двумя фактами, взаимно исключающими друг друга;  $I$  – интерпретатор правил, реализующий процесс вывода. База правил  $R$  и множество целей  $G$  образуют базу знаний.

Логика ПБЗ, определяемая ее И/ИЛИ-графом  $\Gamma$ , может быть задана более наглядно соответствующей  $\Gamma$ -связной логической сетью. Так, например, графу (рис. 1) соответствует логическая сеть, приведенная на рис. 2. Заметим, что функциональный базис логической сети ПБЗ содержит только элементы «И» и «ИЛИ», то есть не является полным.

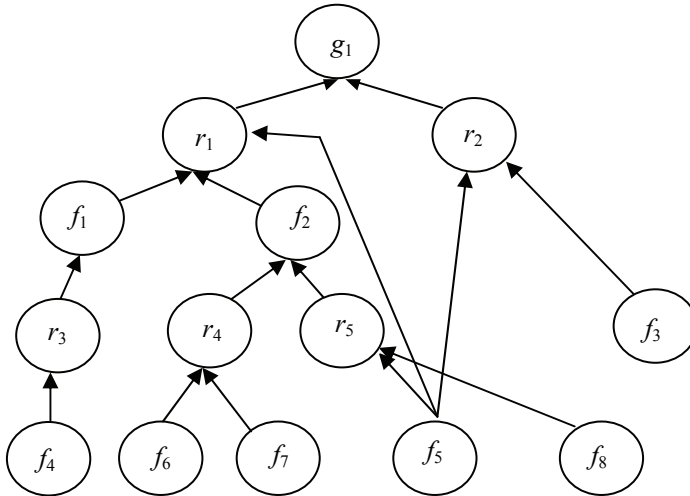


Рис. 1. Пример И/ИЛИ-графа продукционной модели

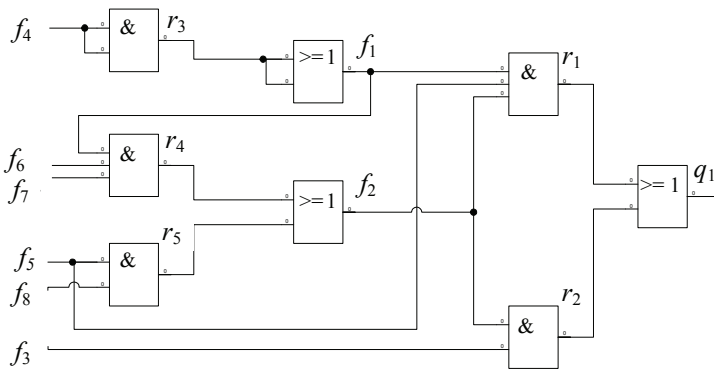


Рис. 2. Эквивалентная И/ИЛИ-графу логическая сеть

В терминах ЛС тестом, обнаруживающим неисправность типа «забывание об исключении», является набор установленных и неустановленных фактов, которые обеспечивают активизацию правила  $r_i$  и активизацию пути в ЛС от линии  $r_i$  до одной из выходных линий, которым соответствуют цели ПБЗ.

Однако преобразование нейронной сети к логической требует, во-первых, дискретизации степени взаимовлияния нейронов; во-вторых, представления значений входных, промежуточных и выходных признаков дискретными значениями 0 или 1; и, в-третьих, прореживания структуры сети путем определения и исключения наименее значимых нейронов и связей. В теории нейронных сетей разработаны методы, позволяющие решать вышеуказанные задачи и представлять нейронные сети в виде наборов правил вида

$$\text{if } (x_1 \text{ or } v_{11} [\text{or } v_{12} \text{ or } \dots]) \text{ and /or } (x_2 \text{ or } v_{21} [\text{or } v_{22} \text{ or } \dots]) \text{ and /or } \dots \text{ then class}_m, \quad (3)$$

где  $x_i$  – входной признак  $i$ ;  $v_{ij}$  – значение  $j$  для признака  $i$ ; or – один из операторов сравнения  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $<>$ ,  $=$ .

Алгоритмы преобразования нейросети к системе продукций можно подразделить на поисковые алгоритмы и алгоритмы, основанные на прямой интерпретации весовых коэффициентов связей. Главным недостатком алгоритмов первого типа является их высокая вычислительная сложность, которая экспоненциально растет при увеличении количества входных и выходных признаков [8]. Исходя из этого, предлагается использовать алгоритмы, предполагающие прямую интерпретацию весовых коэффициентов. Достаточно хорошо зарекомендовал себя алгоритм Glare [8], который предполагает аналитический подход и прямую интерпретацию весовых коэффициентов связей между нейронами входного и промежуточного слоев и связей между нейронами промежуточного и выходного слоя. В пользу выбора Glare также говорят следующие факты:

- набор правил, сгенерированный с помощью такого алгоритма, имеет точность прогноза не меньшую, чем точность прогноза самой нейронной сети;
- отсутствие необходимости модификации метода обучения или структуры самой нейронной сети.

Приведем полный алгоритм генерации тестового множества с использованием алгоритма PODEM [9, 11] для отладки нейронной сети, основанной на трехслойном персептроне.

### Алгоритм генерации тестовых наборов для трехслойного персептрона

#### Входные данные алгоритма

1. Обученная нейросеть  $P$ , основанная на модели трехслойного персептрона. Количество нейронов на первом слое обозначим  $N_1$ , на втором –  $N_2$  и на третьем –  $N_3$ . Активационное значение нейрона  $j$  на слое  $i$  обозначим  $St_{ij}$ . Весовой коэффициент связи нейрона  $j$  на слое  $i$  с нейроном  $k$  слоя  $(i-1)$  обозначим  $W_{ij}^k$ . Для нейронов первого слоя введем дополнительные обозначения:  $m$  – номер входного признака,  $n$  – номер значения входного признака  $m$ , которое кодирует данный нейрон. Таким образом,  $St_{i_1}^{mn}$  обозначает  $i$ -й нейрон первого слоя, соответствующий значению  $n$  атрибута  $m$ .

2. Обучающее множество  $L$ , под которым будем понимать совокупность пар  $\langle X, Y \rangle$  входных и соответствующих выходных векторов системы, где  $X$  – множество входных векторов системы

$$\langle x_1 \dots x_i \rangle_1 \dots \langle x_1 \dots x_i \rangle_n, \quad (4)$$

$Y$  – множество выходных векторов системы  $\langle y_1 \dots y_j \rangle_1 \dots \langle y_1 \dots y_j \rangle_n$ ;  $i$  – размер входных векторов (количество входных признаков),  $i = N1$ ;  $j$  – размер выходных векторов (количество выходных признаков),  $j = N3$ ;  $n$  – мощность обучающего множества (количество обучающих примеров).

#### Выходные данные алгоритма

Тестовое множество  $T$ , совпадающее по структуре с обучающим множеством  $L$  и состоящее из пар входных и соответствующих ожидаемых выходных векторов системы.

#### Алгоритм генерации тестового множества

1. Рассмотреть первый нейрон выходного слоя  $St_{21}$ .

1.1. Составить для нейрона  $St_{21}$  упорядоченную последовательность  $RWIH1$ , состоящую из нейронов входного слоя, упорядоченных по абсолютному значению весового коэффициента связи с нейроном  $St_{21}$

$$RWIH1 = \{ St_{1i}, St_{1j} \dots \mid \text{Abs}(W_{21}^i) < \text{Abs}(W_{21}^{i+1}) \}. \quad (5)$$

1.2. Для нейрона  $St_{21}$  составить последовательность  $RRWIH1$ , полученную путем усечения длины последовательности  $RWIH1$ , полученной в предыдущем шаге. В последовательности  $RRWIH1$  включаются первые  $NWIH$  элементов  $RWIH1$ ,  $1 < NWIH < N2$ .

1.3. Повторить шаги 1.1–1.2 для всех нейронов промежуточного слоя. В результате получить множество последовательностей  $\{RRWIH1, RRWIH2, \dots, RRWIHn\}$ .

1.4. Выделить из обучающего множества  $L$  подмножество  $L1$  такое, что каждый входной вектор  $L1$  активирует первый (рассматриваемый) нейрон выходного слоя

$$L1 \subset L, \forall (\langle X, Y \rangle \subset L1) 1 - Y1 < 1 - Yn, n = 2, \dots, N3. \quad (6)$$

1.5. Для нейрона  $W_{21}^i$  подсчитать индекс значимости  $\Pi_{21}$ . Для этого:

1.5.1. Подать на вход сети элементы обучающего подмножества  $L1$ , подсчитать среднее активационное значение  $AAL_{21}$  нейрона  $St_{21}$ :

$$AAL_{21} = \frac{\sum_{i=n}^1 St_{21}^i}{n}, \quad (7)$$

где  $St_{21}^i$  – активационное значение нейрона при подаче на вход  $P$  примера  $i$  из обучающего подмножества  $T1$ ;  $n$  – мощность подмножества  $T1$ .

1.5.2. Подсчитать индекс значимости  $\Pi_{21}$ :

$$\Pi_{21} = \left| AAL_{21} \times W_{31}^1 \right|. \quad (8)$$

1.6. Повторить шаги 1.5 для каждого нейрона скрытого слоя. Получить последовательность индексов значимостей  $\Pi_{21}, \Pi_{22}, \dots, \Pi_{2, N2}$ .

1.7. Построить последовательность скрытых нейронов  $RWHO_1$  для данного выходного нейрона  $St_{31}$ . Упорядочивание производится по возрастанию соответствующих индексов значимостей, знаки элементов последовательности берутся равными знакам соответствующих весовых коэффициентов связей нейронов скрытого слоя с рассматриваемым нейроном выходного слоя  $St_{31}$ :

$$RWHO_1 = \text{sign}(W_{31}^i) \times |St_{2i}|, \text{sign}(W_{31}^j) \times |St_{2j}| \dots; \Pi_{2i} < \Pi_{2j} \dots \quad (9)$$

1.8. Построить матрицу  $ATTR$ , строками которой являются последовательности  $RRWIN$ , полученные в шагах 1.1–1.3, отсортированные по порядку следования соответствующих элементов последовательности  $RWHO_1$ , полученной в шаге 1.7. При этом инвертировать знаки элементов матрицы  $ATTR$  в тех строках, которые соответствуют отрицательным элементам последовательности  $RRWHO$ :

$$ATTR = \begin{bmatrix} \text{sign}(W_{31}^l) \times RRWIN_l \\ \text{sign}(W_{31}^m) \times RRWIN_m \\ \text{sign}(W_{31}^n) \times RRWIN_n \\ \dots \end{bmatrix}, \quad \Pi_{2l} < \Pi_{2m} < \Pi_{2n} \dots \quad (10)$$

1.9. Построить нулевую матрицу  $RATTR$  размером  $i \times j$ , где  $i$  – количество входных атрибутов;  $j$  – наибольшее количество возможных значений атрибутов.

1.10. Заполнить  $RATTR$ . Для этого:

1.10.1. Рассмотреть элемент  $ATTR_{11}$ . Пусть элементом  $ATTR_{11}$  является нейрон  $St_{1h}^g$ . Тогда  $RATTR_{h,g}$  примет значение  $-1$ , если  $St_{1h}^g < 1$  и  $+1$ , если  $St_{1h}^g > 1$ :

$$RATTR_{h,g} = \text{sign}(St_{1h}^g). \quad (11)$$

1.10.2. Занести пару  $\langle h, g \rangle$  в список  $U$  использованных координат.

1.10.3. Повторить шаг 1.10.1 для  $ATTR_{1,2} \dots ATTR_{1,NWH}$ . При этом если текущие значения пары  $\langle h, g \rangle$  присутствуют в списке использованных координат  $U$ , то  $RATTR_{h,g}$  новое значение не присваивается.

1.10.4. Повторить шаги 1.10.1–1.10.2 для всех строк  $RATTR$ .

1.11. На основе матрицы  $RATTR$  построить классифицирующее правило  $RULE_1$ . Если  $RATTR_{ij} = 1$ , то это означает, что атрибут  $i$  должен принимать  $j$ -е значение, если  $RATTR_{ij} = -1$ , то это означает, что атрибут  $i$  не должен принимать  $j$ -е значение:

$$\begin{array}{l} RULE_1 : \\ \text{if}(St_{1,i}^{m1,n1} = 1(0)) \text{or}(St_{1,j}^{m1,n2} = 1(0) \dots) \text{and} \\ (St_{1,k}^{m2,n1} = 1(0)) \text{or}(St_{1,h}^{m2,n2} = 1(0) \dots) \text{and} \\ \dots \text{then\_classl} \end{array} \quad (12)$$

2. Повторить шаг 1 для всех остальных выходных нейронов  $St_{32}, St_{33}, \dots, St_{3,N3}$ . В итоге получить набор классифицирующих правил вида:

$$\begin{array}{l} \text{if}(St_{1,i}^{m1,n1} = 1(0)) \text{or}(St_{1,j}^{m1,n2} = 1(0) \dots) \text{and} \\ (St_{1,k}^{m2,n1} = 1(0)) \text{or}(St_{1,h}^{m2,n2} = 1(0) \dots) \text{and} \\ \dots \text{then\_classl} \\ \text{if}(St_{1,i}^{m1,n1} = 1(0)) \text{or}(St_{1,j}^{m1,n2} = 1(0) \dots) \text{and} \\ (St_{1,k}^{m2,n1} = 1(0)) \text{or}(St_{1,h}^{m2,n2} = 1(0) \dots) \text{and} \\ \dots \text{then\_class2} \\ \dots \\ \text{if}(St_{1,i}^{m1,n1} = 1(0)) \text{or}(St_{1,j}^{m1,n2} = 1(0) \dots) \text{and} \\ (St_{1,k}^{m2,n1} = 1(0)) \text{or}(St_{1,h}^{m2,n2} = 1(0) \dots) \text{and} \\ \dots \text{then\_classN3} \end{array} \quad (13)$$

3. Для множества правил, полученных в п. 2, построить соответствующую логическую схему  $S$ . Дать имена линиям и элементам схемы. Множество линий обозначить  $L = \{l_1, l_2, \dots, l_m\}$ , множество элементов схемы обозначить  $U = \{u_1, u_2, \dots, u_n\}$ . Составить матрицу инциденций  $M$  размером  $n \times m$ . В матрице  $M$  элемент, находящийся в строке  $i$  и столбце  $j$  равен 1 (-1), если линия  $j$  является входной (выходной) для элемента схемы  $i$ . Если в столбце  $i$  отсутствует значение 1 (-1), то линия  $l_i$  является выходом (входом) схемы. При этом необходимо зафиксировать соответствие между номерами узлов и типами соответствующих узлов (И, ИЛИ).

4. Экспертным методом сформировать множество запрещенных кубов  $C1$ . Множество разрешенных кубов  $C$  можно получить вычитанием из куба всевозможных комбинаций  $R$  значений запрещенных кубов из множества  $C1 : C = R - C1_1 - C1_2 - \dots - C1_h$ .

5. Установить множество тестов  $T$  пустым:  $T = \emptyset$ .

6. Из множества нерассмотренных неисправностей схемы  $S$  выбрать очередную неисправность  $s/a$ . Инициализировать стек  $D$ -границы и внести в него элементы, для которых линия  $s$  является входной. Номера таких элементов соответствуют номерам столбцов ячеек со значением 1 в строке  $s$ . Установить на линии  $s$  значение  $D^a$ . Если же невыбранных неисправностей нет, то завершить алгоритм.

7. Для линии  $s$  определить значения входов, при которых она имеет значение  $D^a$  (выполнить операцию  $D$ -установки). При этом, если присутствует возможность выбора, выбрать путь согласно правилу «а»<sup>1</sup>, остальных кандидатов внести в стек установки  $St$ .

8. Для полученных входных значений определить значения всех элементов, которые однозначно вытекают из полученных входных значений (выполнить операцию импликации).

9. Рассмотреть первый элемент в стеке  $D$ -границы. Выбрать следующий за ним по направлению к выходу элемент и выяснить условия, при которых на его входе установится значение  $D$  (выполнить операцию  $D$ -прохода). Если при таком выборе присутствуют альтернативные элементы, то выбрать элемент согласно правилу «б», остальных же кандидатов внести в стек  $D$ -границы. В случае невозможности  $D$ -прохода для текущего элемента выбрать для его осуществления следующий элемент в стеке  $D$ -прохода. Если же стек  $D$ -прохода пуст, то считать данную неисправность  $s/a$  избыточной и вернуться на шаг 7.

10. Выполнить операцию импликации.

11. Для линий, которые выходят из элементов, для которых определен выход, но еще не определены входы, выяснить значения входа, при котором линия принимает нужное значение (выполнить операцию возврата). В случае появления возможности выбора использовать правило «а». Остальных кандидатов внести в стек возврата  $Back$ .

12. Для полученного в шаге 11 значения входа выполнить операцию импликации. Убедиться в корректности первоначального значения линии (выбранной в шаге 11). Если верное значение не получено, то инвертировать значение входа и повторить попытку. В случае неудачи  $D$ -проход невозможен и необходимо вернуться к шагу 10.

13. Повторить шаги 11–12 для остальных линий в стеке возврата  $Back$ .

14. В случае достижения значением  $D$  выхода схемы  $S$  полученные входные значения считать входным вектором нового теста и вернуться на шаг 6. В против-

---

<sup>1</sup> Под правилом «а» подразумевается правило, согласно которому выбирается ближайшая к входному контакту линия. Правило «б» подразумевает выбор линии, ближайшей к выходному контакту.



ном случае вернуться на шаг 8. Если получен новый тест, то преобразовать его к виду куба  $T1$  и пересечь со множеством разрешенных кубов  $C$  ( $T2 = T1 \cap C$ ).

15. В кубе  $T2$  заменить все значения «X» на «1» и исключить из него повторяющиеся кубы.

16. Объединить полученное множество кубов  $T2$  с уже имеющимся множеством  $T$ , результат присвоить  $T$ :  $T = T + T2$ .

Результатом выполнения данного алгоритма является тестовое множество вида (4). При тестировании для получения эталонных ответов используются эксперты. Несложно убедиться в том, что тестовое множество, которое строится данным алгоритмом, является минимальным. Тесты строятся для набора входных нейронов. Ни один из наборов тестового множества не может быть удален, так как в этом случае тестовое множество, по крайней мере, для одного из значений входных нейронов, окажется не полным.

### Заключение

В настоящей статье описан метод генерации тестового набора, используемый для отладки нейросетевых экспертных систем, основанных на трехслойном персептроне. Метод основан на преобразовании нейронной сети к виду логической схемы посредством выделения из нее продукционных правил и применении метода генерации тестов PODEM, используемого в технической диагностике. Показано, что предложенный метод выявляет ошибки типа «забывание об исключении», а также и ошибки других типов, покрываемые ошибкой «забывание об исключении».

#### Список литературы

1. Уотермен, Д. Руководство по экспертным системам : пер. с англ. / Д. Уотермен. – М. : Мир, 1989. – 388 с.
2. Dayan, P. Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems / P. Dayan, L. Abbott // The MIT Press, Cambridge, Massachusetts, London, England, 2004. – 573 p.
3. Minsky, M. Perceptrons / M. Minsky, S. Papert. – Cambridge : MA : MIT Press, 1969. – 167 p.
4. Rosenblatt, F. Principles of Neurodynamics : Perceptrons and the Theory of Brain Mechanisms / F. Rosenblatt. – New York : Spartan Books, 1962.
5. Долинина, О.Н. Разработка метода тестирования продукционных баз знаний экспертных систем с учетом ошибок типа «забывание об исключении» / О.Н. Долинина. – Саратов, 1997. – 171 с.
6. Поспелов, И.Г. Динамическое описание систем продукции и проверка непротиворечивости продукционных экспертных систем / И.Г. Поспелов, Л.Я. Поспелова. – Изв. АН СССР, ТК, 1987. – № 1. – С. 184–192.
7. Rumelhart, D.E. Learning Representations by Back-Propagating Errors / D.E. Rumelhart, G.E. Hinton, R.J. Williams // Nature. – 1986. – Vol. 323. – P. 533.
8. Gupta, A. Generalized Analytic Rule Extraction for Feedforward Neural Networks / A. Gupta, S. Park, S. Lam // IEEE Transactions on Knowledge and Data Engineering. – 1999. – Vol. 11, No. 6. – P. 985–992.
9. Goel, P. “PODEM – X” : an Automatic Test Generation System for VLSI Logic Structures / P. Goel, B.C. Rosales // Proc. 18-th IEEE Design Automation Conf. – 1981. – Vol. 133. – P. 260–268.
10. Ярмолик, В.Н. Контроль и диагностика цифровых узлов ЭВМ / В.Н. Ярмолик. – М. : Наука и техника, 1988. – 240 с.
11. Киносита, К. Логическое проектирование СБИС : пер. с япон. / К. Киносита, К. Асада, О. Карацу. – М. : Мир, 1988. – 309 с.



## Method of Tests Generating for Debugging of Neuron Networks Expert Systems

O.N. Dolinina, A.K. Kuzmin

*Department «Applied Information Technology»,  
Saratov State Technical University; olga@sstu.ru*

**Key words and phrases:** artificial neuron networks; errors in the database; debugging of neuron networks; method of engineering diagnostics.

**Abstract:** The article presents the key stages of the life cycle of neuron network systems as well as the analysis of errors occurred in the data base of the expert systems. Methods of their debugging are analyzed. The article considers the approach to the task solution of the exhaustive patterns generation, based on the conversion of the knowledge representation structure into the type of the logic network and application of the engineering diagnostics methods. The method of testing of neuron network expert systems based on the spreading of the approach to the model of the laminated perceptron is proposed.

---

## Methode der Testengeneration für Debugging der Neuroexpertensysteme

**Zusammenfassung:** Hier werden Grundetappen des Lebenszyklus der Neuronenetzsysteme, Analyse der Fehler, die in Basen des Wissens der Expertensysteme entstanden, angeführt, auch werden existierende Debuggingmethode analysiert. Hier wird die Behandlung zur Lösung der Aufgabe der Generation der erschöpfenden Testsätze beschrieben, die ist auf Reorganisation der Struktur der Vorstellung des Wissens des Expertensystems zum Art des logischen Netzes und der Anwendung dazu Methode der technischen Diagnose. Auch wird eine Testmethode für das Neuronenetzexpertensystem vorgeschlagen, der ist auf die Verbreitung dieser Behandlung auf das Modell des mehrschichtigen Perzeptrons.

---

## La méthode de la génération des tests pour la mise au point des systèmes d'experts appartenus aux réseaux neuronaux

**Résumé:** Sont présentées les étapes principales du cycle de vie des systèmes neuronaux, l'analyse des erreurs élevées dans les bases de connaissances des systèmes d'experts, sont analysées les méthodes existantes et leur mise au point. Est décrite une approche de traitement du problème de génération des ensembles de tests exhaustifs basée sur la transformation de la structure du système de représentation des connaissances dans les systèmes d'experts pour former un réseau logique, auquel il est possible d'appliquer les méthodes de diagnostic technique. Est proposée une méthode pour tester les systèmes d'experts fondée sur la répartition de cette approche sur le modèle de perceptron multicouche.

---

**Авторы:** *Долинина Ольга Николаевна* – кандидат технических наук, доцент, заведующая кафедрой «Прикладные информационные технологии»; *Кузьмин Алексей Константинович* – ассистент кафедры «Прикладные информационные технологии», ГОУ ВПО «СГТУ».

**Рецензент:** *Большаков Александр Афанасьевич* – доктор технических наук, профессор кафедры «Система искусственного интеллекта», ГОУ ВПО «СГТУ».