

АЛГОРИТМ ИДЕНТИФИКАЦИИ СОСТОЯНИЙ  
МНОГОУРОВНЕВОЙ ТЕХНИЧЕСКОЙ СИСТЕМЫ  
С ИСПОЛЬЗОВАНИЕМ РАСПЛЫВЧАТЫХ КАТЕГОРИЙ  
МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В. Е. Белоусов, И. П. Абросимов, О. В. Губина

Воронежский государственный технический университет

Поступила в редакцию 25.07.2017 г.

**Аннотация.** В статье рассматривается алгоритм извлечения информации для модели представления знаний при идентификации состояний многоуровневой технической системы на основе правил, сформированных лингвистическими переменными с учетом исследуемых классов и минимальной ошибки.

**Ключевые слова:** идентификация, кластер, модель, техническая система, точность, ошибка.

**Annotation.** In article the algorithm of extraction of information for model of representation of knowledge at identification of conditions of multilevel technical system on the basis of the rules created by linguistic variables taking into account the studied classes and the minimum mistake is considered.

**Keywords:** identification, cluster, model, technical system, accuracy, mistake.

Перед исследователями все чаще возникают задачи идентификации состояний в сложных иерархических системах технического с дискретным входом [1], которые охватывают значительные области человеческой деятельности. Допустим, что для некоторой технической системы задано множество входных переменных  $X = \{x_1, x_2, \dots, x_n\}$  и множество дискретных кластеров решений  $D = \{d_1, d_2, \dots, d_m\}$ , которые формируют вектор значений выходной переменной  $y$ . Тогда можно сформировать базу знаний, на основе нечетких правил-продукций с заданными функциями принадлежности. Задача состоит в следующем: заданному вектору входных переменных  $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$  привести в соответствие конкретное решение  $y \in D$  (рис. 1). Чтобы решить указанную задачу необходимо разработать нечеткие продукционные правила, а также алгоритмы вывода

знаний. В результате получена следующая модель [2]:

$$\begin{aligned} &\text{«ЕСЛИ } x_1 = \overline{A_1}(w_1) \text{ И } \dots x_n = \overline{A_n}(w_n) \\ &\text{ТО } y = B \text{» [CF]} \end{aligned} \quad (1)$$

где  $x_i, i = \overline{1, n}$  – входные переменные,  $w_i \in [0, 1]$  – веса условий,  $\overline{A_n}$  – ограничения для четких и нечетких входных условий,  $y$  – результирующая переменная,  $B$  – конкретное значение выхода,  $CF \in [0, 1]$  – достоверность правила.

Для того чтобы организовать логический вывод по условиям (1), зададим обозначения:  $R \in [0, 1]$  – степень реализации относительного участка правила  $R = \min(\mu_{\overline{A_i}}(x_i^*))$ , где  $x_i^*, i = \overline{1, n}$  – конкретные параметры  $n$  входных значений правила,  $\mu_{\overline{A_i}}(x_i^*)$  – функция принадлежности параметров входа  $x_i^*$  к  $\overline{A_i}$ ;  $T \in [0, 1]$  – значений уровня доверия к полученному результату правила  $T = \frac{\sum_{kn=1}^{n_{kn}} w_{kn}}{\sum_{i=1}^n w_i}$ , где  $w_i, i = \overline{1, n}$  – значение весовых функций заданных ограничений  $\overline{A_i}$  на параметры,  $w_{kn}, kn = \overline{1, n_{kn}}$  – веса

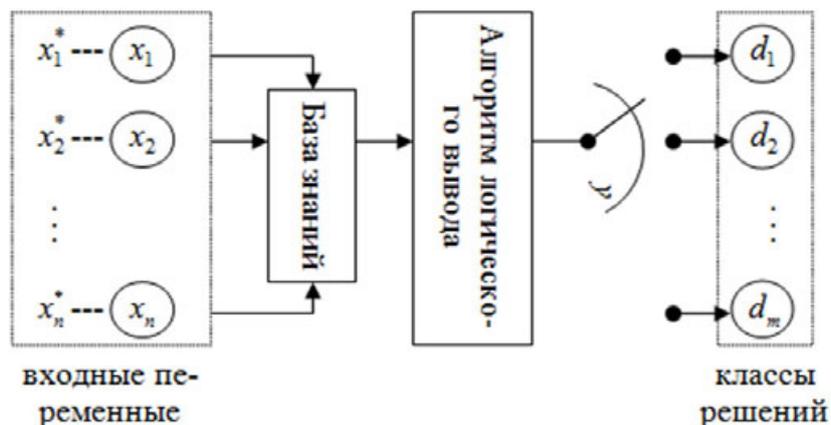


Рис. 1. Схема аппроксимации сложного объекта с дискретным выходом

этих ограничений;  $C = R * T * CF$  – требуемая степень оценки достоверности правила, при заданном  $CF \in [0, 1]$ .

Тогда алгоритм извлечения знаний будет содержать следующие этапы:

**Шаг 1.** Определние входных параметров  $x_i^*$ .

**Шаг 2.** Расчет степени выполнения условной части  $R_r$  для введенного правила  $Rule_r$ ,  $r = 1, N$ .

**Шаг 3.** Определение конфликтного множества, которое включает правила с ненулевой степенью срабатывания:  $S_{conf} = \{Rule_r | R_r \neq 0\}$ .

**Шаг 4.** Для каждого из правил конфликтного множества  $Rule_r \in S_{conf}$  определение веса относительного участка правила  $T_r$ , а также оценки достоверности  $C_r$ .

**Шаг 5.** Этап разрешения конфликта, который состоит из выбора правила с наибольшей оценкой достоверности правила решения:  $Rule_r^* : \max C_r$ .

**Шаг 6.** Определение некоторого значения искомого класса решения  $d_j$  выходного параметра  $y$ .

Представленный алгоритм позволяет классифицировать состояние многоуровневой технической системы, причем повышение точности указанного алгоритма возможно при условии, что параметры исследуемой модели определены с высокой степенью точности, т. е. ошибка минимальна:

$$E = \frac{N_{false}}{N} \rightarrow \min_y, \quad (2)$$

где  $N_{false}$  – число некорректно полученных состояний системы,  $N$  – суммарное значение образов в обучающей выборке.

На рис. 2 представлен пример полносвязной нейронной нечеткой сети. Передача сигнала по сети происходит строго в прямом направлении, от уровня к уровню, слева направо. На выходе из сети формируется сигнал ошибки, распространяющийся в обратном направлении, который определяется на базе функции ошибки [3].

Рассмотрим используемые в модели обозначения: индексы  $i, j, k$  – определяют нейроны сети;  $n$  – итерация  $n$ -го обучающего образа;  $E(n)$  – энергия ошибки на итерации  $n$ ;  $e_j(n)$  – сигнал ошибки на выходе нейрона  $j$  при итерации  $n$ ;  $d_j(n)$  – желаемый отклик нейрона  $j$  для вычисления  $e_j(n)$ ;  $y_j(n)$  – функциональный сигнал;  $w_{ij}(n)$  – синоптический вес;  $\Delta w_{ij}(n)$  – коррекция;  $v_j(n)$  – индуцированное локальное поле  $j$ -го нейрона;  $\phi_j(n)$  – функция активации;  $b_j$  – порог, применяемый к нейрону  $j$  представлен синапсом с весом  $w_{j0} = b_j$ , увязанным с заданным сигналом входа +1;  $x_i(n)$  –  $i$ -й элемент вектора входа;  $o_k(n)$  –  $k$ -й элемент вектора выхода;  $\eta$  – параметр скорости обучения;  $w$  – вектор весов;  $m_l$  – количество узлов слоя  $l$  многослойного персептрона.

Сигнал ошибки нейрона выхода  $j$  при итерации  $n$  определяется:

$$e_j(n) = d_j(n) - y_j(n). \quad (3)$$

Текущий показатель энергии ошибки  $j$ -го нейрона обозначим как  $\frac{1}{2}e_j^2(n)$ . Тогда текущий показатель общей энергии ошибки сети можно найти:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n), \quad (4)$$

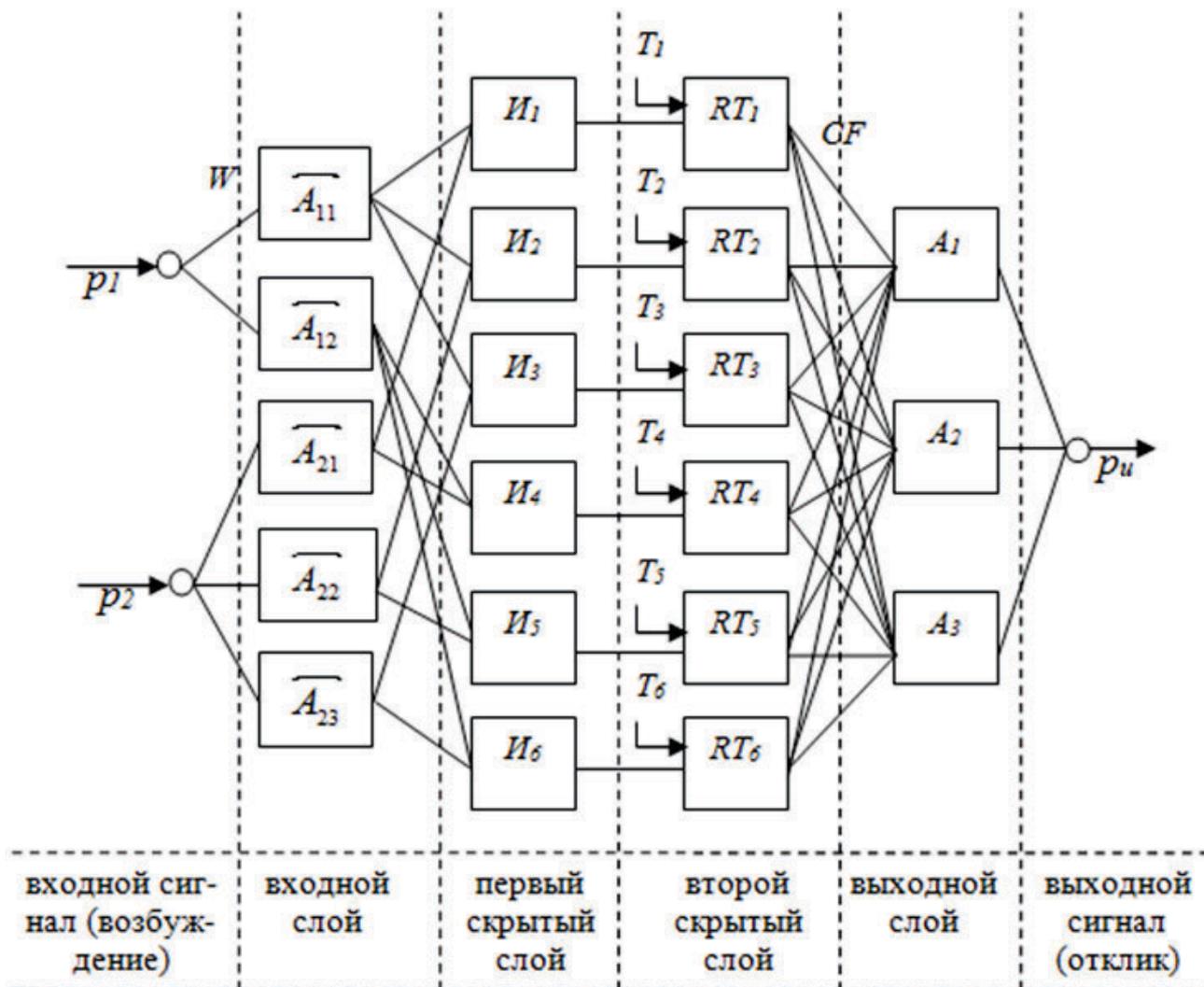


Рис. 2. Примерная структура нейронечеткой сети

Пусть  $N$  – это общее число образов в обучающем множестве. Тогда энергия среднеквадратической ошибки будет вычисляться:

$$E_{av}(n) = \frac{1}{N} \sum_{n=1}^N E(n). \quad (5)$$

Текущая энергия ошибки  $E(n)$ , равно как и средняя энергия ошибки  $E_{av}$ , – это функции синоптических весов и значений порога сети. Для проектируемого обучающего множества энергия  $E_{av}$  является функцией стоимости – мерой эффективности обучения. Настройка весов будет происходить на основании ошибок, вычисленных для каждого образа. Индуцированное локальное поле  $v_j(n)$ , связанное с данным нейроном и полученное на входе функции активации, равно:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n), \quad (6)$$

где  $m$  – общее количество входов (не считая порога) нейрона  $j$ . Синоптический вес  $w_{j0}$  (который соответствует фиксированному входу  $y_0 = +1$ ) эквивалентен порогу  $b_j$ , применяемому к нейрону  $j$ . Функциональный сигнал  $y_j(n)$  на выходе нейрона  $j$  при итерации  $n$  будет равен

$$y_j(n) = \phi_j(v_j(n)). \quad (7)$$

Частная производная  $\partial E(n) / \partial w_{ji}(n)$  является фактором чувствительности [4, 5], функция которого состоит в определении направления поиска в пространстве весов для  $w_{ji}(n)$ :

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}. \quad (8)$$

Продифференцируем обе части (4) по  $e_j(n)$  и получим:

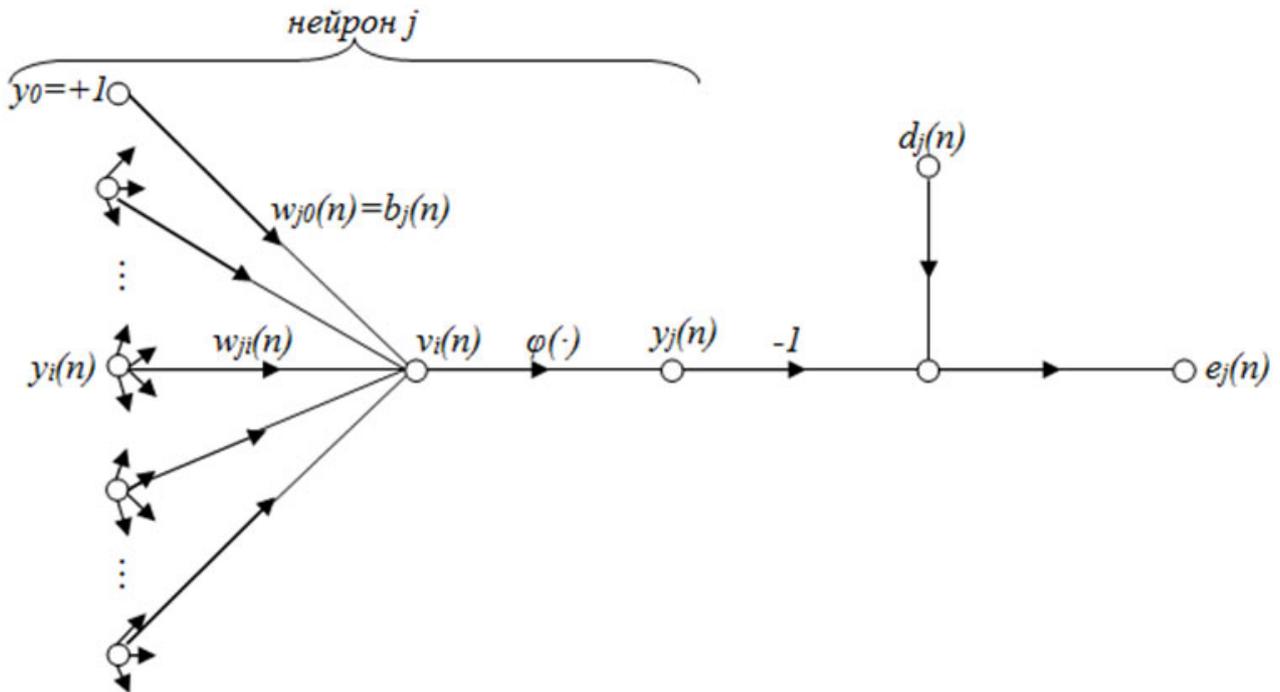


Рис. 3. Граф передачи сигнала в пределах  $j$ -го нейрона

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n). \quad (9)$$

Дифференцируя обе части (3) по  $y_j(n)$ , мы получаем:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1. \quad (10)$$

Дифференцируем (7) по  $v_j(n)$  и получаем:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi'_j(v_j(n)), \quad (11)$$

Далее дифференцируем (6) по  $w_{ji}(n)$ :

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n). \quad (12)$$

Подставляем результаты (9)–(12) в (8), окончательно получаем:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\phi'_j(v_j(n))y_i(n). \quad (13)$$

Коррекция  $\Delta w_{ji}(n)$ , которая применима к  $w_{ji}(n)$ , определяется в соответствии с дельта-правилом:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}, \quad (14)$$

Знак минус в (14) обусловлен реализацией градиентного спуска в весовом пространстве. Соответственно, подставляя (13) в (14), получаем:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \quad (15)$$

где локальный градиент  $\delta_j(n)$  можно определить выражением:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = \quad (16)$$

$$= -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\phi'_j(v_j(n)).$$

Из (15) и (16) можно проследить, что один из главных факторов в вычислении величины коррекции  $\Delta w_{ji}(n)$  весовых коэффициентов – это сигнал ошибки  $e_j(n)$  нейрона  $j$ . В этой связи представляется возможным выделить два различных случая, которые определяются положением нейрона  $j$  в сети.

**Вариант 1.** Нейрон  $j$  – выходной узел. Если нейрон  $j$  находится в выходном слое сети, то для него уже известен соответствующий желаемый отклик. Следовательно, рассчитать сигнал ошибки можно при помощи несложной арифметической операции: на основе (3) рассчитываем сигнал ошибки  $e_j(n)$  (рис. 3) и вычисляем градиент  $\delta_j(n)$  по (16).

**Вариант 2.** Нейрон  $j$  – скрытый узел. Если нейрон  $j$  находится в передаточном слое сети, то желаемый отклик для него неизвестен. Несмотря на то, что скрытый нейрон недоступен, он все равно несет ответствен-

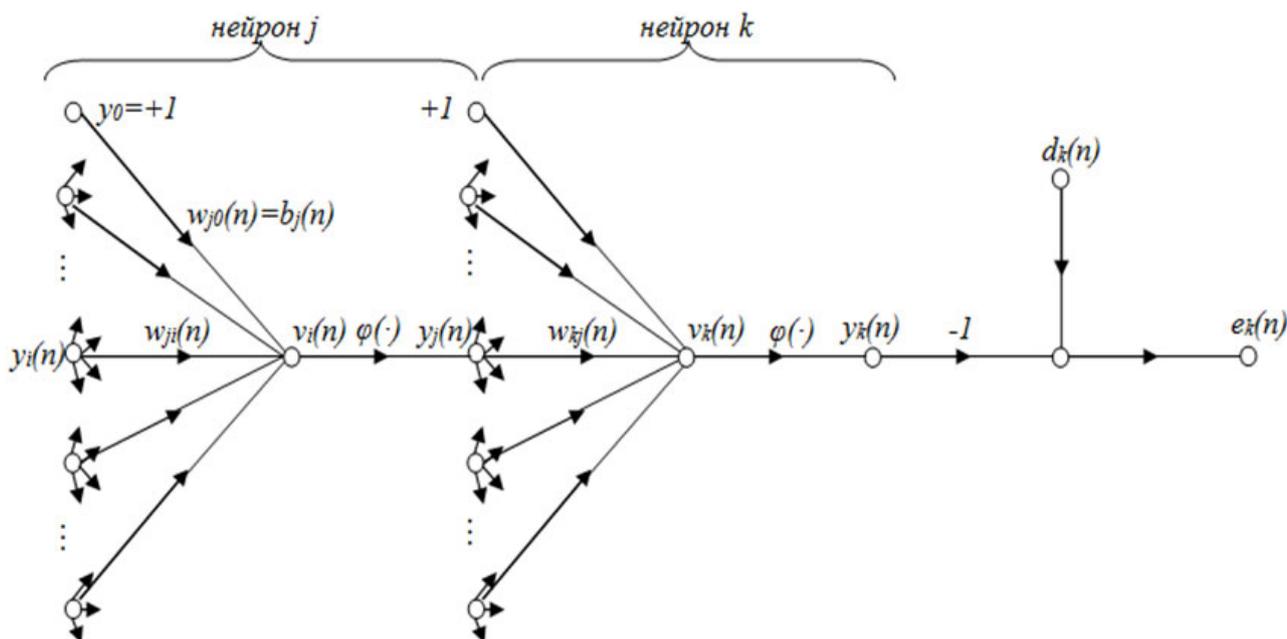


Рис. 4. Граф передачи сигнала с детальным отображением связи нейрона выхода  $k$  со скрытым нейроном  $j$

ность за получаемую на выходе сети ошибку. Проблема заключается в персонализации вклада конкретных скрытых нейронов в общую ошибку и называется задачей назначения коэффициентов доверия. Рассмотрим ситуацию, изображенную на рис. 4, где представлен скрытый нейрон  $j$ . В соответствии с (16) локальный градиент  $\delta_j(n)$  скрытого нейрона  $j$  можно переписать следующим образом:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial y_j(n)} \phi'_j(v_j(n)), \quad (17)$$

где  $j$  – скрытый нейрон; для получения второго результата была применена (11). Для определения частной производной  $\partial E(n) / \partial y_j(n)$  произведем некоторые преобразования. Из рис. 4 видно, что:

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n), \quad (18)$$

где  $k$  – нейрон выхода. Видно, что (18) – это (4), в котором индекс  $j$  заменен на индекс  $k$  (с целью избегания путаницы). Продифференцируем (18) по  $y_j(n)$  и получим:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)}. \quad (19)$$

Теперь к частной производной  $\partial e_k(n) / \partial y_j(n)$  применим правило цепочки и перепишем (19) в эквивалентной форме:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}. \quad (20)$$

Из рис. 4 видно, что для нейрона выхода  $k$ :  $e_k(n) = d_k(n) - y_k(n) = d_k(n) - \phi_j(v_k(n))$ . (21)

Отсюда:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\phi'_k(v_k(n)). \quad (22)$$

Продифференцируем (23) по  $y_j(n)$  и получим:

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n). \quad (24)$$

Подставим (22) и (24) в (20), получим следующее соотношение:

$$\begin{aligned} \frac{\partial E(n)}{\partial y_j(n)} &= -\sum_k e_k(n) \phi'_k(v_k(n)) w_{kj}(n) = \\ &= -\sum_k \delta_k(n) w_{kj}(n), \end{aligned} \quad (25)$$

В итоге, подставляя (25) в (17), мы имеем формулу обратного распространения сигнала ошибки для локального градиента  $\delta_j(n)$  скрытого нейрона  $j$ :

$$\delta_j(n) = \phi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n). \quad (26)$$

Следует отметить, что множитель  $\phi'_j(v_j(n))$  в (26) зависит исключительно от функции активации, которая связана со скрытым нейроном  $j$ . Второй множитель

$\sum_k \delta_k(n)w_{kj}(n)$  взаимосвязан с двумя множествами параметров.

Сведем вместе все соотношения, которые были синтезированы для данного алгоритма:

1. коррекция  $\Delta w_{ji}(n)$ , соединяющая нейроны  $i$  и  $j$ , в отношении синоптического веса, задается следующим правилом дельта:

$$\begin{pmatrix} \text{коррекция} \\ \text{веса} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{параметр} \\ \text{скорости} \\ \text{обучения} \\ \eta \end{pmatrix} \times \begin{pmatrix} \text{локальный} \\ \text{градиент} \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} \text{входной} \\ \text{сигнал} \\ \text{нейрона} \\ y_i(n) \end{pmatrix}$$

– величина локального градиента  $\delta_j(n)$  определяется положением нейрона в сети: если нейрон  $j$  является выходным, то градиент  $\delta_j(n)$  равен произведению производной  $\phi'_j(v_j(n))$  на сигнал ошибки  $e_j(n)$  для нейрона  $j$  (16);

– если нейрон  $j$  является скрытым, то градиент  $\delta_j(n)$  равен произведению производной  $\phi'_j(v_j(n))$  на взвешенную сумму градиентов, которые были вычислены для нейронов следующего скрытого или выходного слоя, непосредственно связанных с данным нейроном  $j$  (26).

Таким образом, разработанный алгоритм вывода знаний на базе нечетких правил про-

дукций позволяет идентифицировать некоторое состояние многоуровневой технической системы в соответствии с заданными кластерами решений, что обеспечивает высокую степень точности идентифицируемого класса состояния.

## СПИСОК ЛИТЕРАТУРЫ

1. Белоусов В. Е., Нгуен Вьет Туан. Алгоритм выбора наилучшего варианта проведения натурального эксперимента в многокритериальных задачах моделирования сложных технических объектов // Системы управления и информационные технологии. – 2016. – №1(63). – С. 55–59.

2. Белоусов В. Е., Царегородцева О. В. Механизмы оценки адекватности процессов идентификации результатов натуральных экспериментов с семантическими моделями представления знаний // Качество продукции: контроль, управление, повышение, планирование: сборник научных трудов Международной молодежной научно-практической конференции (17–18 ноября 2016 года) / редкол.: Павлов Е. В. (отв. ред.); В 2-х томах, Т. 1., Юго-Зап. гос. ун-т., Курск : Из-во ЗАО «Университетская книга», 2016. – С. 94–99.

3. Белоусов В. Е. Моделирование гибких производственных систем с однородным потоком ресурсов / В. Е. Белоусов, О. Ю. Карчевский // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2017. – № 1. – С. 77–81.

**Белоусов Вадим Евгеньевич** – канд. техн. наук, доцент, заведующий кафедрой автоматизации технологических процессов и производств Воронежского государственного технического университета. Тел.: (4732)76-40-07  
E-mail: belousov@vgasu.vrn.ru

**Абросимов Иван Петрович** – аспирант Воронежского государственного технического университета. Тел.: (4732)76-40-07  
E-mail: abrosimov@vgasu.vrn.ru

**Губина Ольга Владимировна** – аспирант Воронежского государственного технического университета. Тел.: (4732)76-40-07  
E-mail: gubina@vgasu.vrn.ru

**Belousov Vadim E.** – candidate of Technical Sciences, head of the department of automation of technological processes and manufactures of the Voronezh state technical university. Tel.: (4732) 76-40-07  
E-mail: belousov@vgasu.vrn.ru

**Abrosimov Ivan P.** – graduate student, Voronezh state technical university. Tel.: (4732) 76-40-07  
E-mail: abrosimov@vgasu.vrn.ru

**Gubina Olga V.** – graduate student, Voronezh state technical university. Tel.: (4732) 76-40-07  
E-mail: gubina@vgasu.vrn.ru