

Субботин С. А.<sup>1</sup>, Благодарев А. Ю.<sup>2</sup>, Гофман Е. А.<sup>3</sup><sup>1</sup>Д-р техн. наук, профессор, заведующий кафедрой программных средств Запорожского национального технического университета, Запорожье, Украина<sup>2</sup>Аспирант кафедры программных средств Запорожского национального технического университета, Запорожье, Украина<sup>3</sup>Канд. техн. наук, старший научный сотрудник научно-исследовательской части Запорожского национального технического университета, Запорожье, Украина

## СИНТЕЗ НЕЙРО-НЕЧЕТКИХ ДИАГНОСТИЧЕСКИХ МОДЕЛЕЙ С ХЭШИРУЮЩИМ ПРЕОБРАЗОВАНИЕМ В ПОСЛЕДОВАТЕЛЬНОМ И ПАРАЛЛЕЛЬНОМ РЕЖИМАХ

**Актуальность.** Решена актуальная задача повышения скорости построения нейро-нечетких моделей по прецедентам.

**Цель работы** – создание метода синтеза нейро-нечетких сетей, обладающего высокой скоростью вычислений и позволяющего реализовать синтез нейро-нечетких сетей в параллельном режиме.

**Метод.** Предложен метод построения нейро-нечетких моделей по прецедентам, который осуществляет сокращение размерности входных данных посредством хэширующего отображения на одномерную ось с сохранением локальной топологии кластеров в признаковом пространстве, оценивает значимость признаков и экземпляров на основе выделенных кластеров, а также формирует разбиение исходного признакового пространства в автоматическом режиме, синтезирует структуру и настраивает параметры нейро-нечеткой модели автоматически, исключая из процесса обучения нейро-нечеткой модели малоинформативные данные, упрощая тем самым структуру получаемых моделей, а также позволяет наиболее вычислительно трудоемкие операции выполнять в параллельном режиме, что позволяет автоматизировать процесс синтеза нейро-нечетких моделей по прецедентам, а также повысить скорость построения нейро-нечетких моделей как при последовательной, так и при параллельной реализации вычислений.

**Результаты.** Разработано программное обеспечение, реализующее предложенный метод, которое использовано при проведении вычислительных экспериментов по исследованию свойств метода.

**Выводы.** Проведенные эксперименты подтвердили работоспособность предложенного метода и реализующего его программного обеспечения, а также позволяют рекомендовать их для применения на практике для решения задач диагностирования и автоматической классификации по признакам.

**Ключевые слова:** нейро-нечеткая сеть, хэш, обучение, синтез, кластер-анализ, диагностика, распознавание.

### НОМЕНКЛАТУРА

ЭВМ – электронная вычислительная машина;

$\varepsilon$  – максимально допустимая ошибка модели;

$\varepsilon_j$  – максимально допустимая погрешность представления значений  $j$ -го признака;

$\mu_{j,l}$  – функция принадлежности к  $l$ -му терму  $i$ -го признака;

$\mu_l$  – функция принадлежности к области  $l$ -го кластера;

$\mu^k$  – функция принадлежности  $k$ -го класса;

$b, c$  – параметры функции принадлежности;

$C^q$  – эталон (центр)  $q$ -го кластера;

$C_j^q$  – координата по  $j$ -му признаку эталона  $q$ -го кластера;

$E$  – ошибка модели;

$f$  – критерий качества модели;

$F()$  – структура нейромодели;

$G$  – номер наибольшего разряда в разрядной сетке ЭВМ;

$i, j$  – номера признаков;

$K$  – число классов;

$L_j$  – число квантований значений  $j$ -го признака;

$l_j^q$  – координата левой границы проекции  $q$ -го кластера на оси  $j$ -го признака;

$N$  – число входных признаков;

$N'$  – число входных признаков после отбора признаков;

$N_C(q, p)$  – число экземпляров  $q$ -го класса в точке  $p$ ;

$N_C(p)$  – число коллизий для точки (разряда)  $p$  при отсечении всех меньших разрядов;

$N_{pr}$  – число параллельных вычислительных модулей, использовавшихся при построении модели;

$opt$  – условное обозначение оптимума;

$p$  – точка;

$p^*$  – точка, соответствующая максимальному разряду хэша, для которой уровень ошибки является приемлемым;

$R(a, b)$  – расстояние между точками  $a$  и  $b$  в пространстве признаков;

$r_j^q$  – координата правой границы проекции  $q$ -го кластера на оси  $j$ -го признака;

$S$  – объем выборки;

$S^*$  – объем редуцированной выборки;

$t$  – время, затраченное на построение нейро-нечеткой модели;

$t_{cp}$  – среднее затрат времени на построение модели;

$w$  – параметры нейромодели;

$x$  – набор входных векторов прецедентов;

$x_*$  – хэш-признак;

$X$  – набор прецедентов;

$x_j$  –  $j$ -й входной признак;

$x^s$  – набор входных признаков  $s$ -го прецедента;

$x_j^s$  – значение  $j$ -го входного  $s$ -го прецедента;

$y$  – выходной признак;

$x_j^{\min}, x_j^{\max}$  – минимальное и максимальное значения признака  $x_j$ ;

$y^s$  – значение выходного признака для  $s$ -го прецедента.

## ВВЕДЕНИЕ

Нейро-нечеткие сети [1–3] являются широко используемым на практике инструментом вычислительного интеллекта, позволяющим строить модели зависимостей по прецедентам в условиях неполной, недостоверной, противоречивой, неточной информации о моделируемой зависимости.

Важными свойствами нейро-нечетких сетей являются их способность извлекать знания из данных в процессе обучения по прецедентам – наблюдениям, точечным реализациям моделируемой зависимости, а также интерпретабельность получаемых моделей [1]. Эти свойства имеют чрезвычайно важное значение в задачах технического и биомедицинского диагностирования, где, с одной стороны, отсутствует или является недостаточным экспертное знание и требуется строить модель на основе наблюдений, а, с другой стороны, необходимо иметь возможность не только принять решение на основе модели, но и понимать процесс принятия решения, то есть обладать интерпретабельной моделью [4].

Еще одним чрезвычайно важным свойством нейро-нечетких сетей является присущий им массивированный параллелизм вычислений [5]. Построенная и обученная аппроксимации зависимости нейро-нечеткая сеть имеет параллельную архитектуру и легко реализуется как в последовательном, так и параллельном режимах вычислений. Поскольку задачи диагностирования во многих практических приложениях предполагают обработку данных, характеризующихся большой размерностью, параллелизм вычислений, имеющийся у нейро-нечетких моделей, способен обеспечить необходимую скорость обработки данных и принятия решений для систем реального времени.

Однако следует отметить, что большинство известных методов синтеза и обучения нейро-нечетких сетей [1–3, 6–9] характеризуются значительными затратами времени на построение моделей и являются последовательными.

Целью данной работы являлось создание метода синтеза нейро-нечетких сетей, обладающего высокой скоростью вычислений и позволяющего реализовать синтез нейро-нечетких сетей в параллельном режиме.

## 1 ПОСТАНОВКА ЗАДАЧИ

Пусть мы имеем исходную выборку  $X = \langle x, y \rangle$  – набор  $S$  прецедентов о зависимости  $y(x)$ ,  $x = \{x^s\}$ ,  $y = \{y^s\}$ ,  $s = 1, 2, \dots, S$ , характеризующихся набором  $N$  входных признаков  $\{x_j\}$ ,  $j = 1, 2, \dots, N$ , и выходным признаком  $y$ . Каждый  $s$ -й прецедент представим как  $\langle x^s, y^s \rangle$ ,  $x^s = \{x_j^s\}$ ,  $y^s \in \{1, 2, \dots, K\}$ ,  $K > 1$ .

Тогда задача синтеза нейромодели зависимости  $y(x)$  будет заключаться в определении такой структуры  $F()$  и значений параметров  $w$  нейромодели, при которых будет удовлетворен критерий качества модели  $f(F(), w, \langle x, y \rangle) \rightarrow \text{ort}$  [4, 5]. Обычно критерий качества обучения нейромодели определяют как функцию ошибки модели:

$$E = \frac{1}{2} \sum_{s=1}^S (y^s - F(w, x^s))^2 \rightarrow \min.$$

Для задач с дискретным выходом ошибку обученной сети можно характеризовать также формулой:

$$E = \frac{100\%}{S} \sum_{s=1}^S |y^s - F(w, x^s)| \rightarrow \min.$$

## 2 ЛИТЕРАТУРНЫЙ ОБЗОР

В настоящее время предложено достаточно много различных архитектур и методов синтеза нейро-нечетких систем [1–3, 5–9]. Однако наиболее удобными для решения задач диагностирования и распознавания образов являются интегрированные нейро-нечеткие системы, представляющие собой прямонаправленные нейронные сети без обратных связей, которые по структуре и функционированию соответствуют системе нечеткого вывода, но при этом могут обучаться как нейронные сети [1, 2].

Наиболее частым подходом к построению интегрированных нейро-нечетких моделей является задание пользователем набора нечетких термов переменных и правил принятия решений, которых отображаются в структуру сети [2]. При этом в процессе обучения, осуществляемого на основе градиентных методов [3, 5] или методов интеллектуального стохастического поиска [10], выполняется настройка параметров термов и весов правил по набору имеющихся наблюдений за моделируемой зависимостью. Достоинствами данного подхода являются его относительная простота, а также интеграция эмпирических наблюдений и человеческих знаний. Недостатком данного подхода является его субъективность, зависимость от человека-эксперта, а также возможная неоптимальность получаемых моделей вследствие избыточности или неполноты выделенных экспертом правил и сформированных термов.

Другим частым подходом к построению интегрированных нейро-нечетких моделей является выполнение кластер-анализа набора прецедентов [1, 2, 11, 12], в результате которого в исходном пространстве признаков выделяются компактные области-кластеры, по центрам которых формируются правила, а также задаются нечеткие термы признаков как проекции кластеров на оси признаков [1]. Достоинством данного подхода является его меньшая зависимость от человека, чем у первого подхода. Недостатками данного подхода являются необходимость задания человеком числа кластеров или параметров методов кластеризации, высокая итеративность кластер-анализа и низкая скорость работы для признаковых пространств большой размерности и выборок данных большого объема.

Для преодоления недостатков рассмотренных выше подходов к построению нейро-нечетких моделей необходимо разработать методы, позволяющие более быстро синтезировать нейро-нечеткие модели, обеспечивая при этом высокий уровень автоматизации путем сокращения вовлечения человека-эксперта в процесс построения модели.

## 3 МАТЕРИАЛЫ И МЕТОДЫ

Поскольку скорость процесса синтеза нейро-нечетких моделей сильно зависит от размерности входа пред-

ставляется целесообразным сократить размерность данных в процессе построения моделей, обеспечивая при этом сохранение локальных свойств расположения кластеров в пространстве признаков. Для этого предлагается использовать хэширующие преобразования [13–17].

В результате хэширующего преобразования исходные данные из  $N$ -мерного пространства отображаются в одномерное, что для больших  $N$  существенно ускоряет выделение кластеров в одномерном пространстве по сравнению с реализацией кластер-анализа в исходном  $N$ -мерном пространстве. Выделенные одномерные кластеры можно использовать для оценки важности экземпляров и признаков. Это позволяет исключить из процесса обучения малоинформативные данные, а, следовательно, существенно ускорить процесс обучения и также упростить структуру получаемых моделей [16].

Поскольку при большом объеме данных и большой размерности входного пространства требуется выполнять значительное число операций обработки данных при построении нейро-нечетких моделей, представляется целесообразным выделить наиболее вычислительно трудоемкие этапы и распараллелить их. Это позволит при использовании параллельных вычислений существенно ускорить процесс построения модели.

Рассмотрим последовательный метод построения нейро-нечетких моделей с хэширующим преобразованием.

Этап инициализации. Для каждого признака  $x_j$ ,  $j = 1, 2, \dots, N$ , определить его минимальное  $x_j^{\min}$  и максимальное  $x_j^{\max}$  значения в выборке  $\langle x, y \rangle$ , а также для каждого  $j$ -го признака задать максимально допустимую погрешность представления значений  $\varepsilon_j$ .

Значение  $\varepsilon_j$  также можно оценить расчетным путем:

$$\varepsilon_j = \min_{\substack{s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ |x_j^x - x_j^p| \mid x_j^x \neq x_j^p \right\}.$$

Затем определить число квантований значений  $j$ -го признака:

$$L_j = \frac{x_j^{\max} - x_j^{\min}}{\varepsilon_j}, j = 1, 2, \dots, N.$$

Этап хэширующего преобразования. Для каждого экземпляра  $\langle x^s, y^s \rangle$  выборки  $\langle x, y \rangle$  определить его хэш  $x^s$ , используя одно из преобразований [13–17].

Упорядочить экземпляры  $\langle x^s, y^s \rangle$  по возрастанию хэша  $x^s$ .

Двигаясь от меньших разрядов хэша к большим,  $p = 0, 1, 2, \dots, G$ , где  $G$  – номер наибольшего разряда в разрядной сетке ЭВМ:

– найти число коллизий для каждой точки (разряда) при отсечении всех меньших разрядов:

$$N_C(p) = \sum_{q=1}^K N_C(q, p) - \max_{q=1,2,\dots,K} \{N_C(q, p)\},$$

где  $N_C(q, p)$  – число экземпляров  $q$ -го класса в точке  $p$ , определяемое по формуле:

$$N_C(q, p) = \sum_{s=1}^S \{1 \mid 2^p(1 + x^s \operatorname{div} 2^p) > x^s \geq 2^p(x^s \operatorname{div} 2^p), y^s = q\};$$

– найти точку, соответствующую максимальному разряду хэша, для которой уровень ошибки является приемлемым:

$$p^* = \max_{p=0,1,\dots,G} \{N_C(p) \mid N_C(p) \leq E\}.$$

Этап редукции признаков. Обрезать хэши всех экземпляров до найденной точки  $p^*$ :

$$x_*^s = x^s \operatorname{div} 2^{p^*}, s = 1, 2, \dots, S.$$

Исключить из исходного набора признаков  $x$  те, которые соответствуют обрезанным частям хэшей. Скорректировать соответствующим образом число признаков  $N$ .

Этап редукции экземпляров. Из каждой группы одинаковых по хэшу экземпляров с одинаковым номером класса оставить только один экземпляр:

$$\langle x_*, y \rangle = \{ \langle x_*, y \rangle \mid \exists p \neq s : x_*^p = x_*^s, y^p = y^s \},$$

после чего определить новое значение числа экземпляров в выборке хэшей  $S^*$ .

Этап формирования четкого разбиения редуцированного исходного пространства признаков. Преобразовать выделенные хэши в области в исходном пространстве признаков, описывая их координатами центра  $C_j^q = \{C_j^q\}$  или координатами границ  $\{ \langle l_j^q, r_j^q \rangle \}$  по каждому  $j$ -му признаку. Если такое преобразование является вычислительно затратным, то начиная с этапа хэширующего преобразования необходимо для каждого хэша запоминать номер экземпляра в исходном пространстве признаков, которому он соответствует. Тогда для заданного хэша можно определить номер экземпляра в исходном пространстве признаков, который и рассматривать как координаты эталона кластера  $C^q$ . Имея набор эталонов  $\{C^q\}$  координаты границ  $q$ -го кластера можно определить как

$$l_j^q = \frac{C_j^q + C_j^p}{2}, p = \arg \min_{\substack{p=1,2,\dots,S^* \\ p \neq q}} \{R(C_j^p, C_j^q) \mid C_j^p \leq C_j^q\};$$

$$r_j^q = \frac{C_j^q + C_j^p}{2}, p = \arg \min_{\substack{p=1,2,\dots,S^* \\ p \neq q}} \{R(C_j^p, C_j^q) \mid C_j^p \geq C_j^q\},$$

$$q = 1, 2, \dots, S^*, j = 1, 2, \dots, N,$$

$$\text{где } R(C^p, C^q) = \sum_{j=1}^N (C_j^p - C_j^q)^2.$$

Этап формирования нечеткого разбиения редуцированного исходного пространства признаков. Сформируем  $\mu_{j,l}(x^s)$  – функции принадлежности к  $l$ -му терму  $i$ -го признака по формуле:

$$\mu_{j,l}(x^s) = e^{-\frac{(x_j^s - b_{j,l})^2}{2c_{j,l}}}.$$

Определим для каждой из областей хэшей (псевдокластеров) функцию принадлежности

$$\mu_l(x^s) = \min_{j=1,2,\dots,N} \{ \mu_{j,l}(x^s) \}.$$

Затем возможно определить функцию принадлежности для каждого из классов:

$$\mu^k(x^s) = \max_{l=1,2,\dots,S^*} \{\mu_l(x^s) | x_*^l \in k\}.$$

Этап структурно-параметрического синтеза нейро-нечеткой модели. Структуру нейро-нечеткой модели предлагается задать на основе классификатора Мамдани [1, 2, 8] – пятислойной нейро-нечеткой сети, изображенной на рис. 1.

На входы сети (первый слой) поступают значения признаков  $\{x_j\}$ , характеризующих распознаваемый экземпляр  $\{x^s\}$ . Узлы второго слоя сети определяют принадлежности экземпляра к нечетким термам признаков  $\{\mu_{j,l}(x^s)\}$ . Узлы третьего слоя определяют принадлежности для каждой из областей хэшей (псевдокластеров)  $\{\mu_l(x^s)\}$ . Узлы четвертого слоя определяют принадлежности для каждого из классов  $\{\mu^k(x^s)\}$ . Единственный узел выходного – пятого слоя определяет четкий номер класса, сопоставляемый распознаваемому экземпляру. Связи между узлами сети формируем на основе правил, в качестве которых можно рассматривать сформированные редуцированные хэши, восстановленные в исходном пространстве признаков:  $\langle x_*^s, x^s, y^s \rangle \Rightarrow x^s \rightarrow y^s$ .

Начальные значения параметров функций принадлежности к термам (параметры блоков второго слоя сети) определим по формулам:

$$b_{j,l} = C_j^l, c_{j,l}^2 = -\frac{(r_j^l - b_{j,l})^2}{2\sqrt{\ln 0,5}}.$$

После задания структуры и начальных значений параметров сети обучим нейро-нечеткую сеть на основе метода эволюционного поиска [10] либо метода обратного распространения ошибки [1].

Схема предложенного метода при последовательной реализации вычислений изображена на рис. 2.

Для ускорения расчетов на основе предложенного метода представляется целесообразным распараллелить вычисления.

На этапе инициализации возможно распараллелить определение минимальных и максимальных значений признаков, определение расстояний между экземплярами по признакам, определение  $\varepsilon_j$  и  $L_j$ .

На этапе хэширующего преобразования хэши экземпляров возможно определять в параллельном режиме. Упорядочивание экземпляров также возможно распараллелить. В параллельном режиме также возможно найти число коллизий для каждой точки (разряда) при отсечении всех меньших разрядов.

На этапе редукции признаков распараллеливанию подлежит обрезка хэшей всех экземпляров до найденной точки  $p^*$ .

На этапе редукции экземпляров распараллеливанию подлежит выделение групп одинаковых хэшей и отбор по одному экземпляру из каждой группы.

На этапе формирования четкого разбиения исходного пространства признаков распараллеливанию подлежит преобразование хэшей в области в исходном пространстве признаков, расчет  $\{C^q = \{C_j^q\}\}$  и  $\langle l_p, r_p \rangle$ .

На этапе формирования нечеткого разбиения исходного пространства признаков возможно в параллельном режиме задать  $\{\mu_{j,l}(x^s)\}$ ,  $\{\mu_l(x^s)\}$  и  $\{\mu^k(x^s)\}$ .

На этапе структурно-параметрического синтеза нейро-нечеткой модели в параллельном режиме можно определить связи между узлами сети, а также начальные значения параметров функций принадлежности к термам. Также возможно в параллельном режиме осуществить обучение нейро-нечеткой модели.

Схема параллельной реализации вычислений этапов предложенного метода изображена на рис. 3.

Как видно из рис. 3, большинство вычислений метода являются параллельными и предполагают расчет одно-

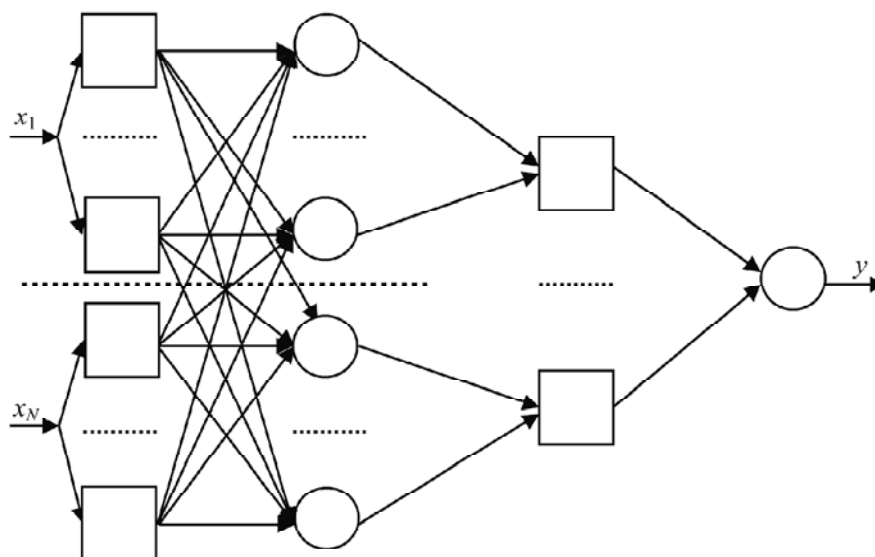


Рисунок 1 – Схема классификатора Мамдани

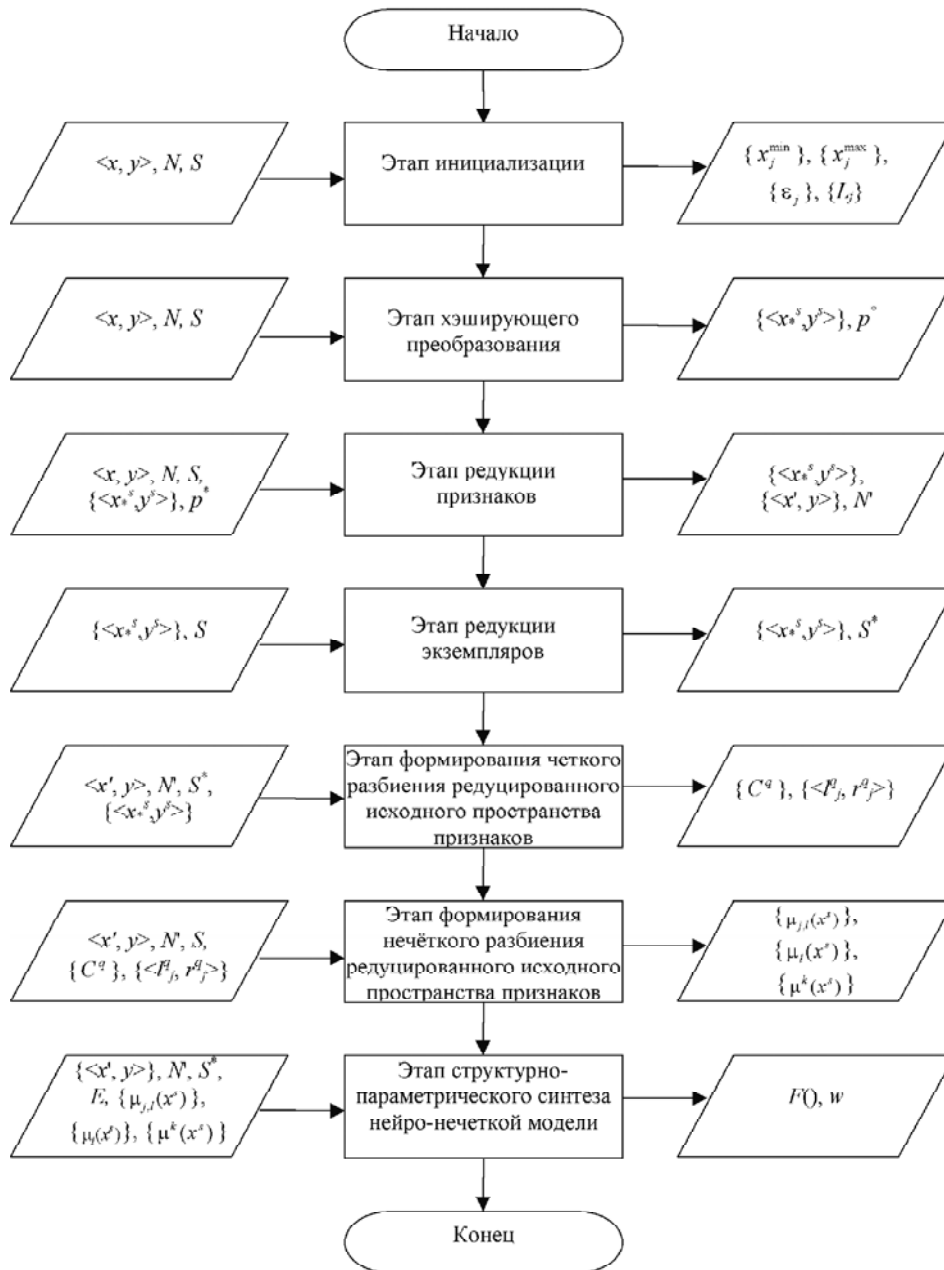


Рисунок 2 – Схема последовательной реализации вычислений

типных величин. Следовательно, предложенный метод обладает массиванным параллелизмом вычислений.

Предложенный метод позволяет решать задачи сокращения размерности выборки данных (отбор информативных признаков и экземпляров, конструирование), кластеризации данных, построения нейро-нечетких моделей по прецедентам.

Также промежуточные результаты, получаемые в процессе выполнения метода, могут быть использованы для оценивания сложности выборок данных. В качестве такой оценки может выступать номер максимального разряда (точки коллизий), дающего приемлемую точность.

Другим вариантом оценивания сложности выборок может служить минимальное число бит для приемлемого по точности хэша выборки. Чем больше требуется

бит для формирования хэша, обеспечивающего приемлемую точность, тем более сложной является выборка данных для заданного хэширующего преобразования.

#### 4 ЭКСПЕРИМЕНТЫ

Для исследования практической применимости предложенного метода он был программно реализован и экспериментально изучен при решении задач построения нейро-нечетких моделей по прецедентам.

Программа, реализующая предложенный метод, позволяет задавать число вычислительных модулей (ядер, процессоров или ЭВМ), работающих в параллельном режиме, из доступных в вычислительной системе.

В табл. 1 приведены характеристики выборок для практических задач [4], использовавшихся при проведении экспериментов.

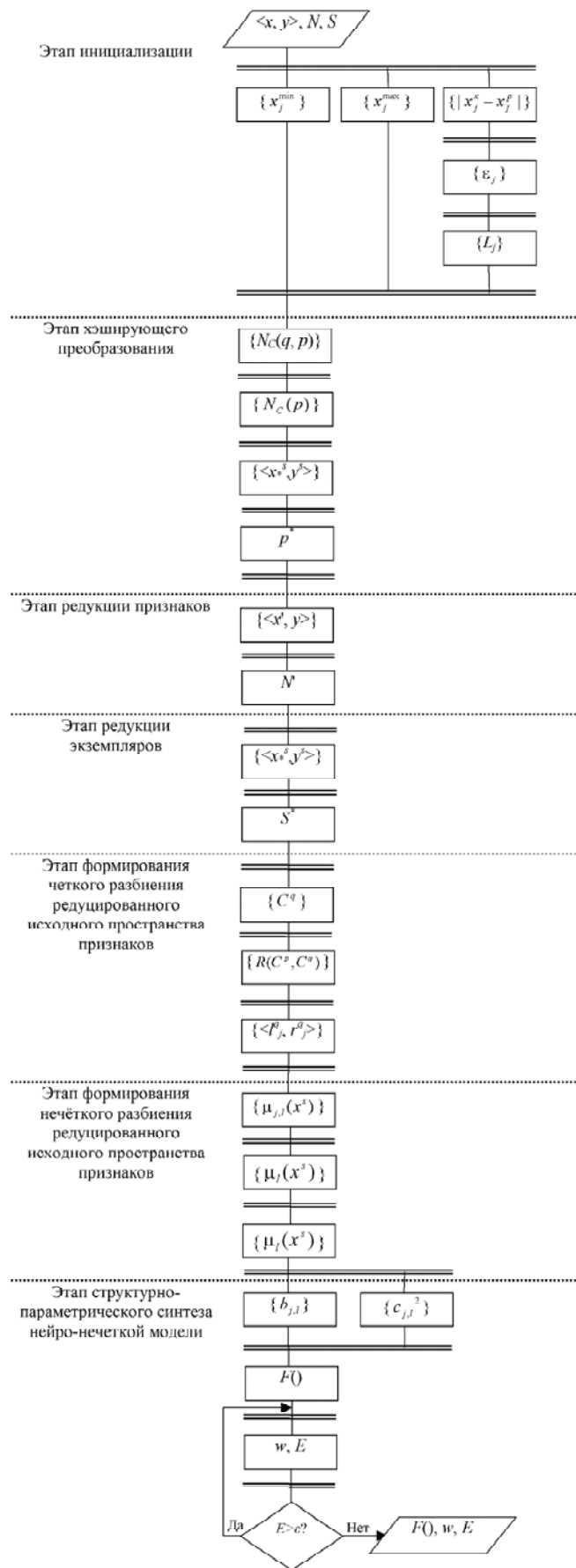


Рисунок 3 – Схема параллельной реализации вычислений этапов метода

Таблица 1 – Характеристики выборок для построения нейро-нечетких моделей

Название задачи	$N$	$S$
Диагностирование лопаток турбины высокого давления газотурбинного авиадвигателя	10240	32
Первичное дифференциальное диагностирование острых воспалений мочевыделительной системы	6	120
Диагностирование рака молочной железы	10	569
Диагностирование хронического обструктивного бронхита	28	205
Диагностирование сердечной аритмии	279	452
Автоматическая классификация ирисов	4	150
Автоматическая классификация автотранспортных средств	26	1062

Для каждой из задач с помощью предложенного метода строились нейро-нечеткие модели при различном числе вычислительных модулей. Для сравнения те же задачи решались на основе оригинальных (нередуцированных) выборок данных, кластер-анализа, сети Мамдани [9] и метода обратного распространения ошибки [1, 2], реализованных в последовательном режиме. Во всех случаях обучение длилось до достижения приемлемой точности.

### 5 РЕЗУЛЬТАТЫ

Эксперименты показали, что базовый метод (нередуцированные данные, кластер-анализ, сеть Мамдани,

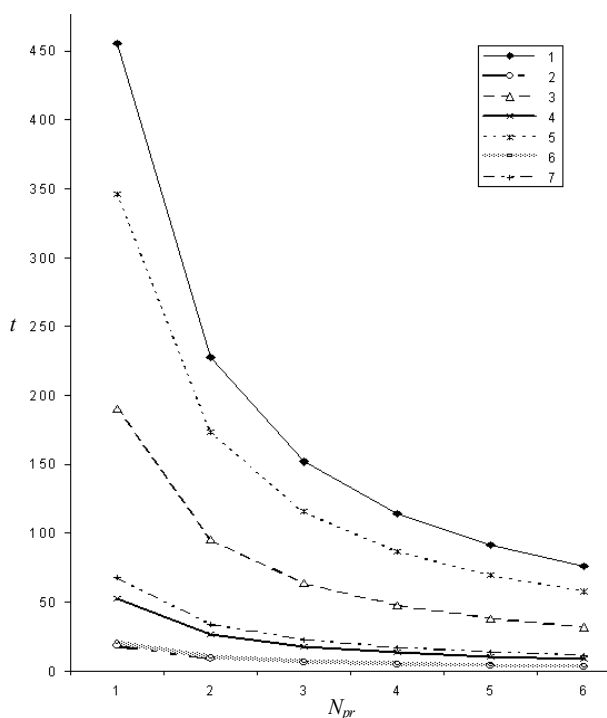


Рисунок 4 – Графики экспериментально полученных зависимостей затрат времени  $t$  на построение нейро-нечеткой модели при разном числе параллельных процессоров  $N_{pr}$  для задач разной размерности: 1 – диагностирование лопаток турбины высокого давления газотурбинного авиадвигателя; 2 – первичное дифференциальное диагностирование острых воспалений мочевыделительной системы; 3 – диагностирование рака молочной железы; 4 – диагностирование хронического обструктивного бронхита; 5 – диагностирование сердечной аритмии; 6 – автоматическая классификация ирисов; 7 – автоматическая классификация автотранспортных средств

метод обратного распространения ошибки в последовательном режиме) в среднем в 5,8 раз требует больше затрат времени по сравнению с предложенным методом при последовательной реализации вычислений (в однопроцессорном режиме).

Результаты проведенных экспериментов по исследованию свойств предложенного метода для различных задач представлены на рис. 4 и рис. 5.

Как видно из рис. 4, с увеличением числа процессоров наблюдается существенное сокращение времени (в разы), затрачиваемого на построение нейро-нечеткой модели на основе предложенного метода. Однако скорость сокращения времени уменьшается по мере увеличения числа используемых процессоров.

Из рис. 5 можно видеть, что рост размерности данных в среднем приводит к росту затрат времени построения модели. При этом увеличение числа используемых процессоров позволяет существенно сократить затраты времени на решение задачи построения модели на основе предложенного метода.

Учитывая широкую практическую доступность в настоящее время двух- и четырехпроцессорных систем для стационарных и мобильных приложений представляется целесообразным рекомендовать использование параллельной реализации предложенного метода. Снижение же порога практической доступности многопроцессорных систем вследствие их ежегодного удешевления с течением времени будет делать предложенный метод еще более востребованным на практике.

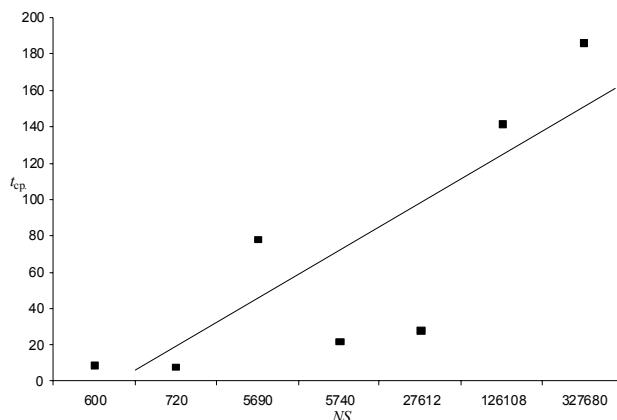


Рисунок 5 – График зависимости средних затрат времени на построение модели  $t_{cp}$  при различном числе процессоров от размерности входа  $NS$  для набора решаемых задач

## 6 ОБСУЖДЕНИЕ

Предложенный метод позволяет в комплексе решать задачи сокращения размерности обучающих данных и синтеза нейро-нечеткой модели в отличие от традиционного подхода, когда редукция данных и синтез модели выполняются на основе различных методов [2, 3, 6, 7]. Предложенный метод отличается от известных методов, в первую очередь, тем, что результаты промежуточных расчетов этапов сокращения размерности используются как априорная информация об обучающей выборке на этапах формирования четкого и нечеткого разбиений признакового пространства, а также при структурно-параметрическом синтезе модели.

Использование предложенного метода будет тем эффективнее, чем больше имеется процессоров или ядер в вычислительной системе, а также чем больше размерность исходной выборки данных.

Отметим, что эффективность метода существенно зависит также от типа выбранного хэширующего преобразования. В случае, когда имеется многопроцессорная вычислительная система и доступное предельное ограничение по времени позволяет, представляется целесообразным использовать набор различных хэширующих преобразований, для каждого из которых синтезировать модель с помощью предложенного метода. Из набора полученных моделей следует выделить те модели, которые обеспечивают приемлемую точность, а из них выбрать как результирующую наиболее простую модель, т.е. содержащую как можно меньше признаков, структурных блоков и параметров.

## ЗАКЛЮЧЕНИЕ

В работе решена актуальная задача повышения скорости построения нейро-нечетких моделей по прецедентам.

Научная новизна полученных результатов состоит в том, что впервые предложен метод построения нейро-нечетких моделей по прецедентам, который осуществляет сокращение размерности входных данных посредством хэширующего отображения на одномерную ось с сохранением локальной топологии кластеров в признаковом пространстве, оценивает значимость признаков и экзemplирует на основе выделенных кластеров, а также формирует разбиение исходного признакового пространства в автоматическом режиме, синтезирует структуру и настраивает параметры нейро-нечеткой модели автоматически, исключая из процесса обучения нейро-нечеткой модели малоинформативные данные, упрощая тем самым структуру получаемых моделей, а позволяет наиболее вычислительно трудоемкие операции выполнять в параллельном режиме, что позволяет автоматизировать процесс синтеза нейро-нечетких моделей по прецедентам, а также повысить скорость построения нейро-нечетких моделей как при последовательной, так и при параллельной реализации вычислений.

Практическая значимость полученных результатов заключается в том, что разработано программное обеспечение, реализующее предложенный метод, которое использовано при проведении вычислительных экспериментов по исследованию свойств метода. Проведенные эксперименты подтвердили работоспособность предложенного

метода и реализующего его программного обеспечения, а также позволяют рекомендовать их для применения на практике для решения задач диагностирования и автоматической классификации по признакам.

Перспективы дальнейших исследований заключаются в том, чтобы разработать и исследовать различные хэширующие преобразования на одномерные оси в аспекте кластеризации данных для построения нейро-нечетких моделей, разработать узкоспецифические более эффективные реализации предложенного метода для конкретных параллельных вычислительных систем, исследовать предложенным методом на более широком классе практических задач диагностирования и автоматической классификации образов.

## БЛАГОДАРНОСТИ

Работа выполнена в рамках государственной научно-исследовательской темы «Методы и средства вычислительного интеллекта и параллельного компьютинга для обработки больших объемов данных в системах диагностирования» (номер гос. регистрации 0116U007419) кафедры программных средств Запорожского национального технического университета при частичной поддержке международного образовательного проекта «Центры передового опыта для молодых ученых» (Ref. No. 544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES) программы «Темпус» Европейского Союза.

## СПИСОК ЛИТЕРАТУРЫ

1. Субботин С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник / С. О. Субботін. – Запоріжжя : ЗНТУ, 2008. – 341 с.
2. Круглов В. В. Нейро-нечеткие методы классификации / В. В. Круглов, О. В. Балашов. – М. : Российский университет кооперации, 2009. – 195 с.
3. Computational intelligence: a methodological introduction / [R. Kruse, C. Borgelt, F. Klawonn et. al.]. – London: Springer-Verlag, 2013. – 488 p. DOI: 10.1007/978-1-4471-5013-8\_1
4. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография / [С. А. Субботин, Ан. А. Олейник, Е. А. Гофман и др.]; под ред. С. А. Субботина. – Харьков : Компания СМІТ, 2012. – 318 с.
5. Субботин С. О. Нейронні мережі : навчальний посібник / С. О. Субботін, А. О. Олійник; під заг. ред. проф. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2014. – 132 с.
6. Гибридные нейро-фаззи модели и мультиагентные технологии в сложных системах: монография / [В. А. Филатов, Е. В. Бодянский, В. Е. Кучеренко и др.]; под общ. ред. Е. В. Бодянского. – Дніпропетровськ : Системні технології, 2008. – 403 с.
7. Buckley J. J. Fuzzy neural networks: a survey / J. J. Buckley, Y. Hayashi // Fuzzy sets and systems. – 1994. – Vol. 66, Issue 1. – P. 1–13.
8. Chai Y. Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application / Y. Chai, L. Jia, Z. Zhang // International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:3, No:3, 2009. – P. 663–670.
9. Mamdani E. H. An experiment in linguistic synthesis with fuzzy logic controller / E. H. Mamdani, S. Assilian // International journal of man-machine studies. – 1975. – Vol. 7, № 1. – P. 1–13.
10. Скобцов Ю. А. Основы эволюционных вычислений / Ю. А. Скобцов. – Донецк: ДонНТУ, 2008. – 330 с.



11. Abonyi J. Cluster analysis for data mining and system identification / J. Abonyi, B. Feil. – Basel : Birkhuser, 2007. – 303 p.
12. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков. – СПб. : БХВ-Петербург, 2003. – 736 с.
13. Super-bit locality-sensitive hashing / [J. Jianqiu, J. Li, Sh. Yany, B. Zhang et al.] // Advances in Neural Information Processing Systems / [eds. P. Bartlett et al.]. – 2012. – Vol. 25. – P. 108–116.
14. Andoni A. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions / A. Andoni, P. Indyk // Communications of the ACM. – 2008. – Vol. 51, No. 1. – P. 117–122.
15. Yang X. A scalable index architecture for supporting multi-dimensional range queries in peer-to-peer networks / X. Yang and Y. Hu // Collaborative computing: networking, applications and worksharing : International conference CollaborateCom-2006, Atlanta 17–20 November 2006 : proceedings. – P. 1–10.
16. Subbotin S. A. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence / S. A. Subbotin // Радиоелектроніка, інформатика, управління. – 2014. – № 1. – С. 120–128.
17. Locality-sensitive hashing scheme based on p-stable distributions / A. Andoni, M. Datar, N. Immorlica, P. Indyk, V. Mirrokni // Nearest neighbor methods in learning and vision: theory and practice / [eds.: T. Darrell, P. Indyk, G. Shakhnarovich]. – MIT Press, 2006. – P. 55–67.

Статья поступила в редакцию 17.10.2016.  
После доработки 18.11.2016.

Субботін С. О.<sup>1</sup>, Благодарьов О. Ю.<sup>2</sup>, Гофман С. О.<sup>3</sup>

<sup>1</sup>Д-р техн. наук, професор, завідувач кафедри програмних засобів Запорізького національного технічного університету, Запоріжжя, Україна

<sup>2</sup>Аспірант кафедри програмних засобів Запорізького національного технічного університету, Запоріжжя, Україна

<sup>3</sup>Канд. техн. наук, старший науковий співробітник науково-дослідної частини Запорізького національного технічного університету, Запоріжжя, Україна

#### СИНТЕЗ НЕЙРО-НЕЧІТКИХ ДІАГНОСТИЧНИХ МОДЕЛЕЙ З ГЕШУВАЛЬНИМ ПЕРЕТВОРЕННЯМ У ПОСЛІДОВНОМУ ТА ПАРАЛЕЛЬНОМУ РЕЖИМАХ

**Актуальність.** Вирішено актуальну задачу підвищення швидкості побудови нейро-нечітких моделей за прецедентами.

**Мета роботи** – створення методу синтезу нейро-нечітких мереж, що має високу швидкість обчислень і дозволяє реалізувати синтез нейро-нечітких мереж у паралельному режимі.

**Метод.** Запропоновано метод побудови нейро-нечітких моделей за прецедентами, що здійснює скорочення розмірності вхідних даних за допомогою гешувального відображення на одновірну вісь зі збереженням локальної топології кластерів у просторі ознак, оцінює значимість ознак і екземплярів на основі виділених кластерів, а також формує розбиття вихідного простору ознак в автоматичному режимі, синтезує структуру і налаштовує параметри нейро-нечіткої моделі автоматично, вилучає з процесу навчання нейро-нечіткої моделі малоінформативні дані, спрощуючи тим самим структуру одержуваних моделей, та дозволяє найбільше обчислювально трудомісткі операції виконувати в паралельному режимі, що дозволяє автоматизувати процес синтезу нейро-нечітких моделей за прецедентами, а також підвищити швидкість побудови нейро-нечітких моделей як при послідовній, так і при паралельній реалізації обчислень.

**Результати.** Розроблено програмне забезпечення, яке реалізує запропонований метод, що використано при проведенні обчислювальних експериментів з дослідження властивостей методу.

**Висновки.** Проведені експерименти підтвердили працездатність запропонованого методу і програмного забезпечення, що його реалізує, а також дозволяють рекомендувати їх для застосування на практиці для рішення задач діагностування й автоматичної класифікації за ознаками.

**Ключові слова:** нейро-нечітка мережа, геш, навчання, синтез, кластер-аналіз, діагностика, розпізнавання.

Subbotin S. A.<sup>1</sup>, Blagodarev A. Yu.<sup>2</sup>, Gofman Ye. A.<sup>3</sup>

<sup>1</sup>Dr. Sc., Professor, Head of the Department of Software Tools of Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine

<sup>2</sup>Post-graduate student of the Department of Software Tools of Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine

<sup>3</sup>PhD, Senior Researcher of Scientific-research Division of Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine

#### THE NEURO-FUZZY DIAGNOSTIC MODEL SYNTHESIS WITH HASHED TRANSFORMATION IN THE SEQUENCE AND PARALLEL MODE

**Context.** The urgent task of improving the speed of neuro-fuzzy model construction by the precedents has been solved.

**Objective** is a creation of a neuro-fuzzy network synthesis method with high speed of computations and allowing to realize the synthesis of neuro-fuzzy networks in parallel mode.

**Method.** The method of neuro-fuzzy model constructing by precedents, which reduces the dimension of the input data by hashing transformation to the one-dimensional axis saving local cluster topology in a feature space, estimates the significance of the features and instances on the basis of selected clusters, and also forms a partition of the original feature space in an automatic mode, synthesizes structure and adjusts parameters of the neuro-fuzzy model automatically, excluding from the training process of the neuro-fuzzy model the uninformative data, thus simplifying the structure of the obtained model, allows to perform most computationally costly operations in parallel mode, that allows to automate the process of neuro-fuzzy model synthesis by precedents, as well as to increase the speed of neuro-fuzzy model construction both in sequential and in parallel implementation of computations.

**Results.** The software implementing proposed method have been developed and used in computational experiments investigating the properties of the method. The experiments confirmed the efficiency of the proposed method and software.

**Conclusions.** The experiments also allow to recommend them for use in practice to solve the problems of diagnosis and automatic classification by the features.

**Keywords:** neuro-fuzzy network, hash, training, synthesis, cluster analysis, diagnosis, recognition.

## REFERENCES

1. Subbotin S. O. Podannya y obrobka znan' u systemakh shtuchnoho intelektu ta pidtrymky pryunyattya rishen' : navch. posibnyk. Zaporizhzhya, ZNTU, 2008, 341 p.
2. Kruglov V. V., Balashov O. V. Neyro-nechetkiye metody klassifikatsii. Moscow, Rossiyskiy universitet kooperatsii, 2009, 195 p.
3. Kruse R., Borgelt C., Klawonn F. et. al. Computational intelligence: a methodological introduction. London, Springer-Verlag, 2013, 488 p. DOI: 10.1007/978-1-4471-5013-8\_1
4. Subbotin S. A., Oleynik An. A., Gofman Ye. A., Zaytsev S. A., Oleynik Al. A. pod red. S. A. Subbotina. Intellektual'nyye informatsionnyye tekhnologii proyektirovaniya avtomatizirovannykh sistem diagnostirovaniya i raspoznavaniya obrazov, monografiya. Khar'kov, Kompaniya SMIT, 2012, 318 p.
5. Subbotin S. O., Oliynyk A. O.; pid zah. red. prof. S. O. Subbotina. Neyronni merezhi : navchal'nyy posibnyk. Zaporizhzhya, ZNTU, 2014, 132 p.
6. Filatov V. A., Bodyanskiy Ye. V., Kucherenko V. Ye. i dr. ; pod obshch. red. Ye. V. Bodyanskogo Gibridnyye neyro-fazzi modeli i mul'tiagentnyye tekhnologii v slozhnykh sistemakh: monografiya. Dnipropeetrovs'k, Sistemni tekhnolognn, 2008, 403 p.
7. Buckley J. J., Hayashi Y. Fuzzy neural networks: a survey, *Fuzzy sets and systems*, 1994, Vol. 66, Issue 1, pp. 1–13.
8. Chai Y., Jia L., Zhang Z. Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol:3, No:3, 2009, pp. 663–670.
9. Mamdani E. H., Assilian S. An experiment in linguistic synthesis with fuzzy logic controller, *International journal of man-machine studies*, 1975, Vol. 7, No. 1, pp. 1–13.
10. Skobtsov Yu. A. Osnovy evolyutsionnykh vychisleniy. Donetsk, DonNTU, 2008, 330 p.
11. Abonyi J., Feil B. Cluster analysis for data mining and system identification. Basel, Birkhuser, 2007, 303 p.
12. Leonenkov A. V. Nepochtoye modelirovaniye v srede MATLAB i fuzzyTECH. Sankt-PeterburgBKHV-Peterburg, 2003, 736 p.
13. Jianqiu J., Li J., Yany Sh., Zhang B. et al. eds. P. Bartlett et al. Super-bit locality-sensitive hashing, *Advances in Neural Information Processing Systems*, 2012, Vol. 25, pp. 108–116.
14. Andoni A., Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, *Communications of the ACM*, 2008, Vol. 51, No. 1, pp. 117–122.
15. Yang X., Hu Y. A scalable index architecture for supporting multi-dimensional range queries in peer-to-peer networks, *Collaborative computing: networking, applications and worksharing : International conference CollaborateCom-2006, Atlanta 17–20 November 2006*, proceedings, pp. 1–10.
16. Subbotin S. A. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence, *Radioelektronika, Computer Science, Control*, 2014, No. 1, pp. 120–128.
17. Andoni A., Datar M., Immorlica N., Indyk P., Mirrokni V. Locality-sensitive hashing scheme based on p-stable distributions, *Nearest neighbor methods in learning and vision: theory and practice* [eds.: T. Darrell, P. Indyk, G. Shakhnarovich]. MIT Press, 2006, pp. 55–67.