УДК 681.3.07

ЗАЙЧЕНКО Ю.П.,
МУРГА Н.А.

ПРИМЕНЕНИЕ СИСТЕМ НА НЕЧЁТКОЙ ЛОГИКЕ К ЗАДАЧЕ МЕДИЦИНСКОЙ ДИАГНОСТИКИ

В данной работе проводится исследование возможности применения нечётких нейронных сетей для задачи медицинской диагностики на примере применения сетей для определения раковых образований в гинекологии. Производится исследование возможностей нечёткой нейронной сети Такаги-Сугено-Канга для решения данной задачи. Так же в работе произведена оценка возможности уменьшения количества тестов, необходимых для процесса диагностирования, что и завершает данную статью.

Investigation of neural network use for medical diagnostic problem on cancer detection example in gynaecology is conducted in this work. Investigation of Takagi-Sugeno-Kang's fuzzy neural network opportunities for this problem solving is conducted. Furthermore, posibility of test's (which are necessary for diagnostics process) number decrease is conducted in this work.

Вступление

Существует очень много работ по применению нейронных сетей в задачах экономики, однако применение нейронных сетей к задаче медицинской диагностики авторам не встречалось. В то же время эта задача весьма актуальна. Применение математических моделей вообще к данной задаче предложено в [1]. Однако ещё авторы данной работы указывают на значительную ошибку, выдаваемую данной моделью и указывают на необходимость поиска более совершенной (модели). Целью данной работы будет анализ применения нечёткой нейронной сети Такаги-Сугено-Канга (TSK-сети) ([2]) к данным, полученным в результате проведения колпоскопического исследовани с целью обнаружения опухолей. Данные представляют из себя (для каждого пациента) 32 теста и 1 диагноз (диагноз – агрегированный диагноз нескольких специалистов). 32 теста рассматриваются как входы сети TSK, а диагноз – выход. Вся выборка данных разбивается на обучающую и проверочную при соотношениях 50:50, 60:40, 70:30, 80:20, 90:10 соответственно. На основании этого строится и обучается нечёткая сеть. Работу завершает анализ возможности уменьшения числа тестов, необходимых для эффективной работы системы.

Постановка задачи

Имеется выборка данных по результатам обследования пациентов. Первые 32 столбца выборки – результаты тестов. Последний столбец – диагноз, поставленный специалистами по результатам тестов. То есть, имеем для каждого случая (пациента) 32 входа и 1 выход. Выборка данных разбивается на обучающую и проверочную в соотношениях: 50:50, 60:40, 70:30, 80:20, 90:10.

В данной работе необходимо провести анализ возможности применения сети TSK для данных обучающих выборок и выполнять проверку эффективности функционирования сети на соответственных проверочных выборках при различном числе правил нечёткого вывода. По результатам экспериментов выделить закономерности и сделать выводы.

Далее, для найденного оптимального соотношения обучающей и проверочной выборки и числа правил произвести последовательное уменьшение тестов, участвующих в диагностике.

1. Применение нечёткой нейронной сети TSK

Ниже приводится описание сети, изложенное в [2].

Обобщённую схему вывода в модели TSK при использовании M правил и N переменных x_j можно представить в следующем виде:

$$R_1 : \ if \quad x_1 \in A_1^{(1)}; x_2 \in A_2^{(1)}; ...; x_n \in A_n^{(1)} \quad then$$

$$y_1 = p_{10} + \sum_{j=1}^{N} p_{1j} x_j ;$$

R_M : *if* $x_1 \in A_1^{(M)}; x_2 \in A_2^{(M)}; ...; x_n \in A_n^{(M)}$ *then* $y_M = p_{M0} + \sum_{j=1}^{N} p_{Mj} x_j$,

где $A_i^{(k)}$ – значение лингвистической переменной x_i для правила R_k с функцией принадлежности $\mu_A^{(k)}(x_i), i = \overline{1,N}; k = \overline{1,M}$.

В нечёткой сети TSK пересечение условий правила R_k определяется функцией принадлежности в форме произведения, то есть:

$$\mu_A^{(k)}(x) = \prod_{j=1}^{N} \mu_A^{(k)}(x_j). \qquad (1)$$

При M правилах вывода композиция выходных результатов сети определяется по следующей формуле (аналогично выводу Сугено):

$$y(x) = \frac{\sum_{k=1}^{M} \omega_k y_k(x)}{\sum_{k=1}^{M} \omega_k}. \qquad (2)$$

где $y_k = p_{k0} + \sum_{j=1}^{N} p_{kj} x_j$. Присутствующие в этом выражении веса ω_k интерпретируются как степень выполнения условий правила: $\omega_k = \mu_A^{(k)}(x)$, которые задаются формулами (1).

Для того чтобы сеть (далее будет иногда заменяться название нечёткая нейронная сеть TSK на, просто, сеть) смогла выполнять функцию, которая от неё ожидается, необходимо её обучить.

Это обучение подразумевает, во-первых, задание количества правил, которые будут использованы сетью TSK (аналогично, сеть и сеть TSK в изложении будут равнозначны). Во-вторых, необходимо задать начальные функции принадлежности (ФП) для каждого входа и каждого правила. В-третьих, задаться видом «то»-части нечётких правил. После выполнения данных действий можно приступать, собственно, к обучению сети.

Возникает немаловажный вопрос: как задать количество правил и ФП входов?

Здесь на помощь приходит ещё один механизм, называемый кластеризацией.

Ниже приведено описание кластеризации, предложенное в [2].

Алгоритм самоорганизации относит вектор *x* к соответственному кластеру данных, которые представляются центром c_i, используя соревновательное обучение.

В данной работе будет интересен такой алгоритм самоорганизации, как алгоритм разностного группирования.

Для его описания используется всё та же работа [2].

Алгоритм разностного группирования – это модификация алгоритма пикового группирования, в котором векторы, подлежащие кластеризации x_j, рассматриваются как потенциальные центры кластеров. Пиковая функция $D(x_i)$ задаётся формулой:

$$D(x_i) = \sum_{j=1}^{N} \exp\left\{ - \frac{\left\| x_i - x_j \right\|^{2b}}{\left(\frac{r_a}{2} \right)^2} \right\}. \qquad (3)$$

где значение коэффициента r_a определяет сферу соседства. На значения $D(x_i)$ значительно влияют только x_j, которые находятся в пределах данной сферы.

При большой плотности точек вокруг x_i значение функции $D(x_i)$ большое. После расчёта значений пиковой функции для каждой точки x_i, выбирается вектор x, для которого мера плотности $D(x)$ окажется самой большой. Именно эта точка и становится первым центром c_1.

Выбор следующего центра c_2 возможен после исключения предыдущего центра и всех точек, лежащих в его окрестности.

Пиковая функция переопределяется следующим образом:

$$D_{new}(x_i) = D(x_i) - D(c_1) \cdot \exp\left\{ - \frac{\left\| x_i - c_1 \right\|^{2b}}{\left(\frac{r_b}{2} \right)^2} \right\}. \qquad (4)$$

При новом определении функции *D* коэффициенты r_b обозначают новые значения константы, которая задаёт сферу соседства очередного центра. Обычно придерживаются условия, что $r_b \geq r_a$.

После модификации значения пиковой функции ищется новая точка *x*, для которой

$D_{new}(x) \to \max$. Она становится новым центром.

Процесс поиска очередного центра возобновляется после исключения всех компонент, которые отвечают уже отобранным точкам. Инициализация завершается в момент фиксации всех центров, которые предусмотрены начальными условиями.

В соответствии с описанным алгоритмом происходит самоорганизация множества векторов x, которая состоит в нахождении оптимальных значений центров, которые представляют множество данных с минимальной погрешностью.

Если мы имеем дело с множеством учебных данных в виде пар векторов $(x_i \quad d_i)$, так как это имеет место при обучении с учителем, то для нахождения центров, которые отвечают множеству векторов d_i, достаточно сформировать расширенный вектор: $[x_i, d_i] \to x_i$.

Процесс группирования, который проводится с использованием расширенных векторов x_i, позволяет определить также и расширенные версии центров c_i.

С учётом того, что размерность каждого нового центра равна сумме размерностей векторов x и d, то в описании этого центра можно выделить часть p, соответствующую вектору x (первые N компонент) и остаток q, который отвечает вектору d. Таким образом, можно получить центры как входных переменных, так и ожидаемых выходных значений $c_i = [p_i, q_i], i = \overline{1, K}$.

Теперь необходимо на вопрос: как именно может помочь данный алгоритм для решения поставленной проблемы?

Ответ очевиден. Количество полученных кластеров после его применения и будет искомым количеством правил. Значения компонент центров кластеров будут определять центры ФП. Дисперсии ФП (берутся ФП гауссовского вида) задаются некоторыми начальными значениями, которые в дальнейшем будут корректироваться.

«Если»-часть правил построена.

Рассматривается «то»-часть.

А здесь возникает два сценария дальнейших действий: считать функцию в «то»-части либо линейной, либо константой.

Необходимо разобраться с каждым из этих случаев в отдельности.

Для начала, константа. Как задать данную константу для каждого правила? А данная задача уже решена. Выше описано применение алгоритма разностного группирования для обучения с учителем. Так вот, размерность вектора d для данного случая равна единице. Выходит, для каждого i-го правила q_i и будет искомой константой.

Сеть можно «доучить» (подкорректировать дисперсии), используя метод Back Propagation, с которым можно ознакомиться в [2].

Далее, линейная функция. Для её построения можно применить метод наименьших квадратов (МНК далее). Необходимо рассмотреть, как это можно сделать, однако, вместе с алгоритмом, который будет использоваться для обучения сети такого вида – гибридным алгоритмом обучения нечётких нейронных сетей.

Приводится выдержка из его описания, предложенного в работе [2].

В гибридном алгоритме параметры, которые подлежат адаптации, разделяются на две группы. Первая из них состоит из линейных параметров p_{kj} – «то»-части нечётких правил, а вторая – из параметров нелинейных ФП «если»-части. Уточнение параметров происходит в два этапа.

На первом этапе при фиксации отдельных значений параметров функций принадлежности (в первом цикле – это значения, которые получены путём инициализации), решая систему линейных уравнений, рассчитываются линейные параметры p_{kj} полинома TSK. При известных значениях ФП зависимость для выхода можно представить в виде линейной формы относительно параметров p_{kj}:

$$y = \sum_{k=1}^{M} \omega'_k \left(p_{k0} + \sum_{j=1}^{N} p_{kj} x_j \right). \qquad (5)$$

где

$$\omega'_k = \frac{\prod_{j=1}^{N} \mu_A^{(k)}(x_j)}{\sum_{r=1}^{M} \prod_{j=1}^{N} \mu_A^{(r)}(x_j)}, \; k = \overline{1, M}. \qquad (6)$$

При размерности обучающей выборки $L\left(x^{(l)}, d^{(l)}\right)$, $(l = \overline{1, L})$ и замене выходного сиг-

нала сети ожидаемым значением $d^{(l)}$, получаем систему из L линейных уравнений вида:

$$\begin{bmatrix} \omega'_{11} & \omega'_{11}\cdot x_1^{(1)} & \dots & \omega'_{11}\cdot x_N^{(1)} & \dots & \omega'_{1M} & \omega'_{1M}\cdot x_1^{(1)} & \dots & \omega'_{1M}\cdot x_N^{(1)} \\ \omega'_{21} & \omega'_{21}\cdot x_1^{(2)} & \dots & \omega'_{21}\cdot x_N^{(2)} & \dots & \omega'_{2M} & \omega'_{2M}\cdot x_1^{(2)} & \dots & \omega'_{2M}\cdot x_N^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \omega'_{L1} & \omega'_{L1}\cdot x_1^{(L)} & \dots & \omega'_{L1}\cdot x_N^{(L)} & \dots & \omega'_{LM} & \omega'_{LM}\cdot x_1^{(L)} & \dots & \omega'_{LM}\cdot x_N^{(L)} \end{bmatrix} \bullet \begin{bmatrix} p_{10} \\ p_{11} \\ \dots \\ p_{1N} \\ \dots \\ p_{M0} \\ p_{M1} \\ \dots \\ p_{MN} \end{bmatrix} = \begin{bmatrix} d^{(1)} \\ d^{(2)} \\ \dots \\ d^{(L)} \end{bmatrix} \qquad (7)$$

где ω'_{lj} означает уровень активации (вес) условия i-го правила при предъявлении l-го входного вектора $x^{(l)}$. Это выражение можно записать в матричном виде:

$$Ap = d \qquad (7)$$

Размерность матрицы A равна $L\cdot(N+1)\cdot M$. При этом количество рядов L обычно бывает значительно больше количества столбцов $(N+1)\cdot M$. Для отыскания p применим метод наименьших квадратов.

На втором этапе после фиксации линейных параметров p_{kj} рассчитываются фактические выходные сигналы $y^{(l)}, l=\overline{1,L}$, для чего используется линейная зависимость:

$$y = Ap \qquad (8)$$

После этого рассчитывается вектор ошибки $\varepsilon = y - d$ и критерий:

$$E = \frac{1}{2}\sum_{l=1}^{L}\left(y\left(x^{(l)}\right) - d^{(l)}\right)^2 \qquad (9)$$

Сигналы ошибок распространяются через сеть в обратном направлении соответственно к методу Back Propagation, прямо к первому слою сети (ФП), где могут быть рассчитаны компоненты вектора градиента целевой функции относительно параметров ФП. После вычисления вектора градиента делается шаг спуска градиентным методом.

После уточнения нелинейных параметров снова запускается процесс адаптации линейных параметров функции TSK (первый этап) и не нелинейных (второй этап). Этот цикл повторяется до тех пор, пока стабилизируются все параметры процесса.

2. Уменьшение количества входных переменных для диагностики

Теперь необходимо ответить ещё на один вопрос: действительно для диагностики нужны все входные переменные или их число можно уменьшить.

Необходимо определиться с тем, по какому признаку будут исключаться тесты. Здесь ответ, с одной стороны, прост и очевиден, а с другой – очень сложен. Прост он тем, что порядок исключения будет определяться величиной корреляции результатов тестов. Сложность возникает в том, что если есть пара элементов, для которых коэффициент корреляции равен 0,98001 и пара элементов, для которых коэффициент корреляции равен 0,98, то сказать точно между какими элементами больше зависимость – нельзя, так как последующие эксперименты могут существенно изменить картину. Однако в данной работе считается, что чем больше корреляция, тем строго больше зависимость между элементами и рассмотрение данной проблемы оставлено для последующих работ.

Строится корреляционная матрица для всех тестов – A, где a_{ij} – коэффициент корреляции между тестами i и j.

Далее, допускается, что найден максимальный элемент в матрице A – a_{ij}.

Выбирается из i-го и j-го (произвольно) какой тест оставляется для рассмотрения, а второй – исключается.

Допускается, что исключается тест под номером i.

Из матрицы A вычёркивается строка и столбец с индексом i.

Далее, необходимо определиться с критерием, не выполнение которого будет свидетельствовать о том, что текущее количество тестов недостаточно для постановки диагноза.

Допустима ошибка диагностики 0,005. Следовательно, на проверочной выборке должно выполнятся условие

$\max_i \left| y_i - \overline{y_i} \right| < 0{,}005$, где y_i – значение, поставленное врачами; $\overline{y_i}$ – диагноз, выданный системой. Принимается данное выражение за вышеуказанный критерий и обозначается как K.

3. Результаты экспериментов

Теперь можно перейти к практическому рассмотрению указанных выше вопросов.

Прежде всего, производится, как было сказано ранее, кластеризация и, одновременно, строится сеть с константой в «то»-части.

Все данные, которые имеются (193 пациента) отображаются в единичный гиперкуб. $r_a = r_b$. И в таблице будет обозначено RI (range influence).

Количество полученных правил в таблице будет обозначено RQ (rules quantity).

Критерием эффективности работы алгоритма берётся $RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{y_i}\right)^2}$.

Дисперсии для гаусовских функций принадлежности для каждого входа будут свои и одинаковые.

Результаты экспериментов заносятся в нижеприведённые таблицы. Отметим отдельно, что строка «Об:Пров» обозначает отношение обучающей выборки к проверочной.

Таблица 1

Об:Пров	50:50			60:40		
RI	**RQ**	**RMSE train**	**RMSE test**	**RQ**	**RMSE train**	**RMSE test**
0,1	64	$1{,}8315\cdot10^{-17}$	0,0332	64	$1{,}5072\cdot10^{-17}$	0,0414
0,2	64	$2{,}6832\cdot10^{-17}$	0,0312	64	$3{,}1898\cdot10^{-17}$	0,0389
0,3	64	$3{,}2746\cdot10^{-17}$	0,0297	64	$3{,}7296\cdot10^{-17}$	0,0371
0,4	64	$5{,}9620\cdot10^{-17}$	0,0282	64	$4{,}5869\cdot10^{-17}$	0,0351
0,5	62	$6{,}0560\cdot10^{-17}$	0,0277	61	$5{,}4587\cdot10^{-17}$	0,036
0,6	55	$2{,}3649\cdot10^{-16}$	0,0623	57	$3{,}6669\cdot10^{-16}$	0,0387
0,7	47	$2{,}1801\cdot10^{-16}$	0,0552	47	$1{,}6546\cdot10^{-16}$	0,047
0,8	32	$4{,}3250\cdot10^{-16}$	0,0376	33	$2{,}2483\cdot10^{-16}$	0,0637
0,9	21	$2{,}7109\cdot10^{-16}$	0,0504	27	$4{,}2174\cdot10^{-16}$	0,0776
1	19	$6{,}3821\cdot10^{-16}$	0,0549	20	$7{,}7099\cdot10^{-16}$	0,0912

Таблица 2

Об:Пров	70:30			80:20		
RI	**RQ**	**RMSE train**	**RMSE test**	**RQ**	**RMSE train**	**RMSE test**
0,1	70	$1{,}4754\cdot10^{-17}$	0,0435	89	$1{,}2581\cdot10^{-17}$	0,0331
0,2	70	$2{,}1681\cdot10^{-17}$	0,0418	89	$2{,}0689\cdot10^{-17}$	0,0302
0,3	70	$3{,}3746\cdot10^{-17}$	0,0388	89	$2{,}9218\cdot10^{-17}$	0,0293
0,4	69	$4{,}7760\cdot10^{-17}$	0,0365	88	$4{,}1680\cdot10^{-17}$	0,0286
0,5	68	$5{,}2207\cdot10^{-17}$	0,0358	85	$5{,}0317\cdot10^{-17}$	0,0269
0,6	64	$5{,}6841\cdot10^{-17}$	0,0386	74	$8{,}3585\cdot10^{-17}$	0,0290
0,7	51	$2{,}0565\cdot10^{-16}$	0,0534	57	$2{,}3107\cdot10^{-16}$	0,047
0,8	37	$4{,}0681\cdot10^{-16}$	0,0967	34	$6{,}5636\cdot10^{-16}$	0,0956
0,9	28	$4{,}5956\cdot10^{-15}$	0,1212	25	$9{,}5112\cdot10^{-16}$	0,0865
1	6	$1{,}2788\cdot10^{-15}$	0,1691	5	$2{,}3033\cdot10^{-15}$	0,1637

Учебная выборка – 90%, проверочная – 10%.

Таблица 3

RI	RQ	RMSE train	RMSE test
0,1	96	$1{,}1710\cdot10^{-17}$	$2{,}9216\cdot10^{-18}$
0,2	96	$2{,}1647\cdot10^{-17}$	$5{,}9195\cdot10^{-17}$
0,3	96	$2{,}6863\cdot10^{-17}$	$8{,}1233\cdot10^{-17}$
0,4	96	$3{,}3784\cdot10^{-17}$	$9{,}6381\cdot10^{-17}$
0,5	92	$4{,}4979\cdot10^{-17}$	$1{,}3867\cdot10^{-16}$
0,6	80	$1{,}6441\cdot10^{-16}$	$3{,}0583\cdot10^{-16}$
0,7	61	$2{,}2497\cdot10^{-16}$	$4{,}5655\cdot10^{-16}$
0,8	44	$5{,}5951\cdot10^{-16}$	$1{,}3725\cdot10^{-15}$
0,9	31	$1{,}0292\cdot10^{-15}$	$3{,}4575\cdot10^{-15}$
1	5	$1{,}7631\cdot10^{-15}$	$3{,}3432\cdot10^{-15}$

Таким образом, видно, что данный алгоритм позволяет построить очень хорошую модель диагностирования. Также видно, что качество диагностики возрастает ростом числа правил (падением индекса влияния). Однако, как видно из графиков (Рис. 2, Рис. 3), этот рост монотонным в общем случае назвать нельзя.

Если же характеризовать зависимость «рост качества диагностики – рост объёма обучающей выборки», то следует более внимательно присмотреться к графикам:

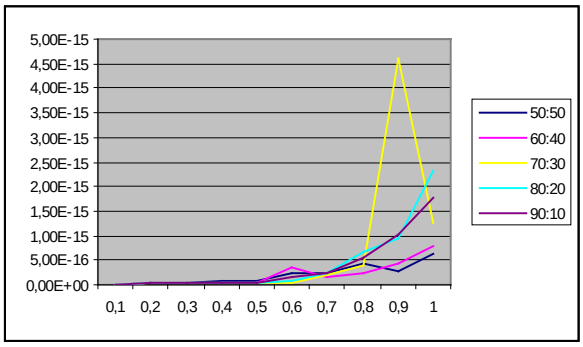

Рис. 1 Зависимость RI – RMSE train для таблиц 1 – 3

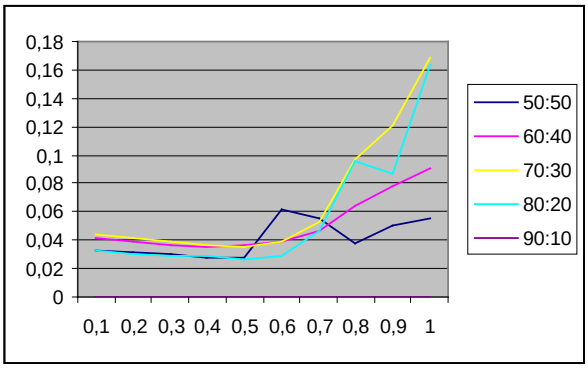

Рис. 2 Зависимость RI – RMSE test для таблиц 1 – 3

Как видно из графиков на рис. 1, для обучающей выборки при $RI = 0{,}1 \div 0{,}5$ будет точно наблюдаться прямая зависимость между качеством диагностики и числом правил нечёткого вывода, а для остальных *RI* четкой зависимости не наблюдается; а для проверочной (рис. 2) – возрастание ошибки до соотношения выборок 7:3 и дальнейшее стремительное падение оной.

Далее рассматривается случай, когда функция в «то»-части линейна.

Задаются следующие начальные условия. Правила и функции принадлежности берутся из предыдущих экспериментов (кластеризации). Обозначения в таблицах аналогичны (хотя RI уже не актуален). Обучение будет проводиться гибридным методом.

Таблица 4

Об:Пров	**50:50**			**60:40**		
RI	**RQ**	**RMSE train**	**RMSE test**	**RQ**	**RMSE train**	**RMSE test**
0,1	64	$2{,}3976 \cdot 10^{-8}$	0,033091	64	$1{,}7003 \cdot 10^{-8}$	0,041256
0,2	64	$2{,}3981 \cdot 10^{-8}$	0,031066	64	$1{,}7008 \cdot 10^{-8}$	0,038731
0,3	64	$2{,}411 \cdot 10^{-8}$	0,029487	64	$1{,}7095 \cdot 10^{-8}$	0,036763
0,4	64	$2{,}5551 \cdot 10^{-8}$	0,027863	64	$1{,}8073 \cdot 10^{-8}$	0,034738
0,5	62	$2{,}235 \cdot 10^{-6}$	0,027419	61	$1{,}6779 \cdot 10^{-6}$	0,035122
0,6	55	$5{,}51 \cdot 10^{-6}$	0,028531	57	$3{,}931 \cdot 10^{-6}$	0,03407
0,7	47	$8{,}112 \cdot 10^{-6}$	0,030635	47	$6{,}314 \cdot 10^{-6}$	0,043796
0,8	32	$8{,}5395 \cdot 10^{-6}$	0,03862	33	$7{,}1297 \cdot 10^{-6}$	0,041018
0,9	21	$1{,}9289 \cdot 10^{-5}$	0,038845	27	$1{,}0663 \cdot 10^{-5}$	0,046861
1	19	$3{,}6898 \cdot 10^{-5}$	0,045584	20	$2{,}8545 \cdot 10^{-5}$	0,047587

Таблица 5

Об:Пров	**70:30**			**80:20**		
RI	**RQ**	**RMSE train**	**RMSE test**	**RQ**	**RMSE train**	**RMSE test**
0,1	70	$1{,}4761 \cdot 10^{-8}$	0,043158	89	$1{,}4916 \cdot 10^{-8}$	0,033651
0,2	70	$1{,}4767 \cdot 10^{-8}$	0,041357	89	$1{,}4919 \cdot 10^{-8}$	0,030443
0,3	70	$1{,}4817 \cdot 10^{-8}$	0,038156	89	$1{,}4985 \cdot 10^{-8}$	0,0293
0,4	69	$5{,}7658 \cdot 10^{-8}$	0,035841	88	$5{,}5153 \cdot 10^{-8}$	0,0288
0,5	68	$1{,}3224 \cdot 10^{-6}$	0,035571	85	$6{,}0647 \cdot 10^{-8}$	0,0291
0,6	64	$3{,}4702 \cdot 10^{-6}$	0,043127	74	$3{,}7547 \cdot 10^{-6}$	0,031
0,7	51	$4{,}9418 \cdot 10^{-6}$	0,047605	57	$6{,}8552 \cdot 10^{-6}$	0,0359
0,8	37	$7{,}3412 \cdot 10^{-6}$	0,048571	34	$1{,}0559 \cdot 10^{-5}$	0,0148
0,9	28	$1{,}0643 \cdot 10^{-5}$	0,046314	25	$1{,}7685 \cdot 10^{-5}$	0,0376
1	6	0,00022524	0,095813	5	$6{,}9307 \cdot 10^{-4}$	0,1178

Обучающая выборка – 90%, проверочная – 10%.

Таблица 6

RI	RQ	RMSE train	RMSE test
0,1	96	$1{,}3748\cdot10^{-8}$	$7{,}3358\cdot10^{-8}$
0,2	96	$1{,}3750\cdot10^{-8}$	$7{,}3361\cdot10^{-8}$
0,3	96	$1{,}3806\cdot10^{-8}$	$7{,}3317\cdot10^{-8}$
0,4	96	$1{,}4470\cdot10^{-8}$	$7{,}3233\cdot10^{-8}$
0,5	92	$5{,}4135\cdot10^{-7}$	$1{,}3909\cdot10^{-6}$
0,6	80	$3{,}2039\cdot10^{-6}$	$9{,}8092\cdot10^{-6}$
0,7	61	$6{,}0106\cdot10^{-6}$	$2{,}4068\cdot10^{-5}$
0,8	44	$9{,}8820\cdot10^{-6}$	$3{,}0978\cdot10^{-5}$
0,9	31	$1{,}6437\cdot10^{-5}$	$6{,}3682\cdot10^{-5}$
1	5	$7{,}0728\cdot10^{-4}$	0,003

Как видно, тенденции здесь аналогичны экспериментам с алгоритмом разностного группирования. Наблюдается зависимость роста качества диагностики с ростом числа правил. Что касается зависимости между объёмом выборки и качеством диагностики, то можно сказать следующее.

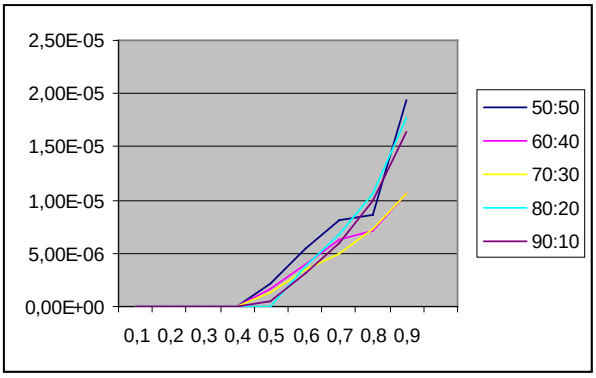

Рис. 3 Зависимость RI – RMSE train для таблиц 4 – 6

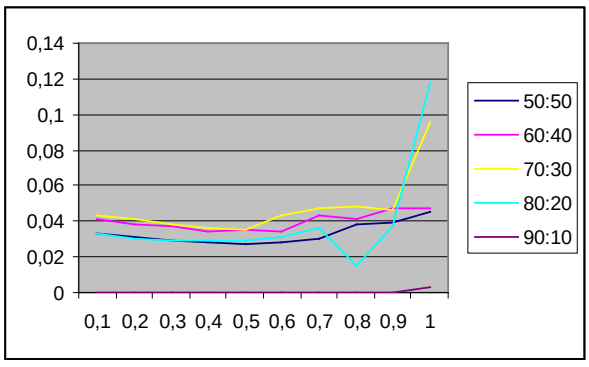

Рис. 4 Зависимость RI – RMSE test для таблиц 4 – 6

На обучающей выборке (рис. 3) эта зависимость не имеет постоянный характер и колеблется. Для проверочной выборки (рис. 4) наблюдается возрастание ошибки с ростом объёма обучающей выборки до соотношения 7:3 и дальнейшее стремительное падение ошибки.

4. Сравнительный анализ алгоритмов

Красноречивее всех описаний говорят результаты, зафиксированные в таблицах 1-3 и 4-6. При функциях-константах в «то»-части правил, полученных после кластеризации система выдавала гораздо лучшие результаты и быстрее шли вычисления, чем линейные функции, получаемые с помощью МНК (см. описание гибридного алгоритма). Необходимо отметить также, что применение для этой же задачи чёткой нейронной сети Back Propagation, рассмотренное в [3], показало, что она по параметру качество диагностики уступает нечёткой сети TSK в обоих случаях. А что касается параметра скорость обучения – уступает случаю функции-константы в «то»-части нечётких правил, но превосходит случай линейной функции.

Однако, как показывают эксперименты, у алгоритмов есть и общие свойства. Из опытов видно (таблицы 1-6), что с ростом числа правил растёт точность диагностики, как для алгоритма разностного группирования (функция-константа), так и гибридного алгоритма (линейная функция). Для обоих алгоритмов наблюдается зависимость падения точности диагностики на проверочной выборке до соотношения выборок 7:3 и дальнейший стремительный рост точности. Однако, что касается обучающей выборки, то по поводу точности диагностики можно сказать, что с ростом объёма растёт точность диагностики для случая функции-константы в «то»-части правил при $RI = 0{,}1 \div 0{,}5$, а про второй случай – чёткой закономерности не наблюдается.

5. Уменьшение числа тестов, необходимых для постановки диагноза

К диагностическим данным применяются инструкции (пункт 2), предложенные ранее

для решения вопроса о количестве тестов, необходимых для диагностики.

Берётся отношение обучающей выборки к проверочной 9:1. Для формирования системы диагностики применяется алгоритм разностного группирования. Обозначается через *n* – число тестов, которые используются для диагностики. RI берётся равным 0,1. *K* – критерий, приведённый во втором пункте.

Результаты занесены в таблицу 7.

Таблица 7

n	RQ	RMSE train	RMSE test	K
2	76	$9{,}1466\cdot10^{-15}$	$2{,}5951\cdot10^{-14}$	$3{,}6710\cdot10^{-13}$
3	88	$9{,}8243\cdot10^{-16}$	$3{,}5033\cdot10^{-15}$	$6{,}1173\cdot10^{-14}$
4	95	$2{,}9052\cdot10^{-17}$	$7{,}9815\cdot10^{-17}$	$7{,}7716\cdot10^{-16}$
5	95	$3{,}2477\cdot10^{-17}$	$7{,}6187\cdot10^{-17}$	$6{,}6613\cdot10^{-16}$
6	96	$2{,}1861\cdot10^{-17}$	$6{,}4230\cdot10^{-17}$	$6{,}6613\cdot10^{-16}$
7	96	$2{,}0654\cdot10^{-17}$	$6{,}5948\cdot10^{-17}$	$7{,}77168\cdot10^{-16}$
8	96	$3{,}5415\cdot10^{-17}$	$6{,}5741\cdot10^{-17}$	$6{,}6613\cdot10^{-16}$
9	96	$1{,}7497\cdot10^{-17}$	$3{,}9496\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
10	96	$1{,}7743\cdot10^{-17}$	$6{,}2081\cdot10^{-17}$	$8{,}8818\cdot10^{-16}$
11	96	$1{,}5131\cdot10^{-17}$	$3{,}8456\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
12	96	$2{,}0499\cdot10^{-17}$	$5{,}4912\cdot10^{-17}$	$6{,}6613\cdot10^{-16}$
13	96	$1{,}6495\cdot10^{-17}$	$6{,}60624\cdot10^{-17}$	$6{,}6613\cdot10^{-16}$
14	96	$1{,}8812\cdot10^{-17}$	$4{,}3088\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
15	96	$1{,}5806\cdot10^{-17}$	$3{,}1263\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
16	96	$1{,}4932\cdot10^{-17}$	$3{,}5029\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
17	96	$1{,}7960\cdot10^{-17}$	$3{,}9873\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
18	96	$1{,}4720\cdot10^{-17}$	$3{,}3978\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
19	96	$1{,}7069\cdot10^{-17}$	$3{,}5090\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
20	96	$1{,}2195\cdot10^{-17}$	$3{,}0116\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
21	96	$1{,}4985\cdot10^{-17}$	$3{,}0080\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
22	96	$1{,}4824\cdot10^{-17}$	$3{,}5931\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
23	96	$1{,}5059\cdot10^{-17}$	$2{,}8923\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
24	96	$1{,}3239\cdot10^{-17}$	$3{,}2038\cdot10^{-17}$	$4{,}4409\cdot10^{-16}$
25	96	$1{,}2587\cdot10^{-17}$	$2{,}3005\cdot10^{-17}$	$2{,}2204\cdot10^{-16}$
26	96	$1{,}5864\cdot10^{-17}$	$2{,}2263\cdot10^{-17}$	$2{,}2204\cdot10^{-16}$
27	96	$1{,}3964\cdot10^{-17}$	$2{,}9947\cdot10^{-17}$	$3{,}3307\cdot10^{-16}$
28	96	$1{,}6132\cdot10^{-17}$	$1{,}7530\cdot10^{-17}$	$2{,}2204\cdot10^{-16}$
29	96	$1{,}2920\cdot10^{-17}$	$1{,}4681\cdot10^{-17}$	$2{,}2204\cdot10^{-16}$
30	96	$1{,}2898\cdot10^{-17}$	$2{,}9216\cdot10^{-18}$	$5{,}5511\cdot10^{-17}$
31	96	$1{,}1404\cdot10^{-17}$	$2{,}9216\cdot10^{-18}$	$5{,}5511\cdot10^{-17}$

Таким образом, видно, что во всех опытах значение критерия *K* удовлетворяется и, следовательно, количество тестов можно значительно уменьшить.

Проводились ещё опыты для различных значений RI с целью добиться наименьшего числа правил при наименьшем числе тестов. Все опыты здесь приведены не будут, но вот несколько примеров:

9 тестов:

Таблица 8

RI	RQ	RMSE train	RMSE check	K
1	2	0,0233	0,0672	0,5364
0,6	11	$5{,}6436\cdot10^{-15}$	$1{,}0704\cdot10^{-14}$	$1{,}2401\cdot10^{-13}$

7 тестов:

Таблица 9

RI	RQ	RMSE train	RMSE check	K
1	3	0,0218	0,0621	0,6588
0,7	6	0,0164	0,0647	0,7306
0,5	32	$5{,}784\cdot10^{-15}$	$1{,}922\cdot10^{-14}$	$1{,}1446\cdot10^{-13}$

Эксперименты показали, что сокращать количество тестов можно. Однако не следует считать, что так же легко их можно уменьшить на практике. Во-первых, врачу необходимо обезопасить себя от поломки оборудования, которое производит тест. Во-вторых, есть заболевания, для которых значения тестов могут быть одинаковые и, поэтому, необходимо проводить дополнительные тесты для определения вида заболевания (в данном случае – действительно ли это рак, или фоновое заболевание). В-третьих, очевидно, что человек не может оперировать одновременно таким большим числом информации, которое, де-факто, требуют от него результаты экспериментов. Естественно в мозгу врача-диагноста происходят несколько отличные процессы от модели. И т.д.

Выводы

В данной работе был проведён анализ применения нечёткой нейронной сети TSK к задаче медицинской. Было проведено ряд вычислительных экспериментов, в результате которых были установлены некоторые факты. Первое, нечёткая нейронная сеть Такаги-Сугено-Канга применима для рассматриваемой задачи. Второе, для задачи, рассматриваемой в работе, применение функции-константы в «то»-части нечётких правил оказалось более эффективным, чем применение линейной функции. Третье, наблюдалась чёткая зависимость роста ошибки на проверочной выборке от соотношения выборок 50:50 до 70:30 в обоих случаях, в то же время на обучающей выборке никакие закономерности не просматривались за исключением случая функции-константы в «то»-части правила и только для $RI = \overline{0{,}1;0{,}5}$. Четвёртое, количество входов системы (тестов) можно значительно сократить. Однако на практике это не стоит делать по указанным в статье причинам. Полученные результаты подтверждают эффективность применения нечёткой нейронной сети TSK к задаче медицинской диагностики, однако ряд рассуждений показывают, что оптимальная система для решения данной задачи не найдена. Поиск данной системы и составит предмет последующих работ.

Список литературы

1. Steve Saggese, Trevor Johnson, Ivy Basco, Yalew Tamrat. Automatic Segmentation of Uterine Cervix for in vivo localization and Identification of Cervical Intraepithelial Neoplasia.– Apogen Technologies 7545 Metropolitan Dr San Diego, CA 92108.
2. Зайченко Ю.П. Основи проектування інтелектуальних систем. Навчальний посібник. – К.: Видавничий Дім «Слово», 2004. – 352 с.
3. Мурга Н.А. Применение нейросетей при колпоскопическом осмотре шейки матки с целью обнаружения кансероподобных образований. Системний аналіз та інформаційні технології: Матеріали X Міжнародної науково-технічної конференції (20-24 травня 2008 р., Київ).–К.:НТУУ «КПІ» 2008.– 235 сторінка.