УДК 004.93.51

Т.М. КОСОВСКАЯ

САМООБУЧАЮЩАЯСЯ СЕТЬ С ЯЧЕЙКАМИ, РЕАЛИЗУЮЩИМИ ПРЕДИКАТНЫЕ ФОРМУЛЫ

Косовская Т.М. **Самообучающаяся сеть с ячейками, реализующими предикатные формулы.**

Аннотация. Рассматривается модель перенастраиваемой сети с ячейками, реализующими предикатные формулы, имеющие вид элементарных конъюнкций. В отличие от классических нейронных сетей предлагаемая модель имеет два блока: блок обучения и блок решения. При ошибках, возникающих при использовании блока решения, подключается блок обучения. Кроме того, конфигурация сети не фиксируется заранее, а меняется каждый раз после работы блока обучения. Базой для создания перенастраиваемой логико-предикатной сети является логико-предметный подход к решению задач искусственного интеллекта, а также понятие неполной выводимости предикатной формулы, позволяющее выделять общие подформулы элементарных конъюнкций.

Ключевые слова: искусственный интеллект, формула исчисления предикатов, уровневое описание классов, самообучающаяся распознающая сеть.

Kosovskaya T.M. **Self-training Network with the Sells Implementing Predicate Formulas.**

Abstract. A model of self-modificated predicate network with cells implementing predicate formulas in the form of elementary conjunction is suggested. Unlike a classical neuron network the proposed model has two blocks: a training block and a recognition block. If a recognition block has a mistake then the control is transfered to a training block. Always after a training block implementation the configuration of a recognition block is changed. The base of the proposed logic-predicate network is a logic-objective approach to AI problems solving and level description of classes as well as the notion of partial deducibility which allows to extract common sub-formulas of elementary conjunctions.

Keywords: artificial intelligence, pattern recognition, predicate calculus formulas, level description of a class, self-training recognition network

---

1. Введение. Традиционно при моделировании задач искусственного интеллекта (ИИ), а особенно задач распознавания образов, исследуемый объект рассматривается как неделимое целое и описывается глобальными признаками, характеризующими его свойства. Такой подход плохо приспособлен к моделированию сложных объектов, характеризующихся свойствами его элементов и отношениями между ними.

40 лет назад появилось большое количество монографий с одним и тем же названием «Искусственный интеллект» (среди них, например, книга с другим названием [1]), в которых предлагалось использование языка исчисления предикатов (ИП) и автоматического доказательства теорем с помощью метода резолюций для решения разнообразных за-

дач этой тематики. Язык ИП [2] вполне адекватен для моделирования сложных и изменяющихся объектов. Однако в этих монографиях не были сделаны оценки числа шагов решения задач в такой модели, что не позволило применять её на практике. В 2006 году вышел перевод монографии почти с таким же названием [10], в которой вновь предлагается использование языка ИП. Из оценок числа шагов приводится лишь экспоненциальная зависимость длины описания объекта в виде строки некоторых значений от его описания на языке ИП.

В работе автора [4] доказаны оценки числа шагов алгоритмов, решающих задачи ИИ при их моделировании с помощью языка ИП. Анализ этих оценок позволил разработать иерархические многоуровневые описания [6] целевых условий, существенно уменьшающие время решения задач. На основе многоуровневых описаний в [5] было предложено построение нейронной сети. Однако на тот момент методика обучения такой сети не была разработана. Возможность автоматического создания логико-предметной сети по обучающей выборке появилась после разработки алгоритма построения многоуровневого описания классов [9].

Широко распространена модель, в которой элемент классической искусственной нейронной сети [3] представляет из себя сумматор взвешенных входов, после которого находится передаточная функция, приводящая значение выхода сумматора в промежуток $[0, 1]$. Конфигурация нейронной сети заранее фиксируется и в процессе обучения меняются только значения весов входов сумматора.

Ниже предлагается модель логико-предикатной нейронной сети, имеющей два блока: блок обучения и блок распознавания. Каждый из блоков в качестве своих элементов имеет предикатную формулу в виде элементарной конъюнкции. Входами элемента сети являются значения предметных переменных для соответствующей элементарной конъюнкции и значения атомарных предикатных формул, задающих свойства предметных переменных и отношения между ними.

Конфигурация блока обучения формируется в процессе обучения сети. После предварительного обучения в этом блоке определяется конфигурация блока распознавания. Блок обучения — это «долго работающий» блок. В отличие от него блок распознавания — это «быстро работающий» блок. Несмотря на то, что блок обучения работает действительно долго (решается NP-трудная задача), это соответствует тому, что человек обучается годами, чтобы потом решать многие задачи в течение секунд.

2. Общая постановка задачи. Пусть исследуемый объект представлен как множество своих элементов $\omega = \{\omega_1, \ldots, \omega_t\}$. На ω задан набор предикатов $p_1, \ldots, p_n$, характеризующих свойства элементов ω и отношения между ними. Логическим описанием $S(\omega)$ объекта ω называется множество всех атомарных формул или их отрицаний, истинных на ω. Множество всех объектов разбито на классы $\Omega = \cup_{k=1}^{K} \Omega_k$. Логическим описанием класса Ω_k называется формула $A_k(\overline{x})$, заданная в виде дизъюнкции элементарных конъюнкций, такая что если $A_k(\overline{\omega})$ истинна, то $\omega \in \Omega_k$.[1]

С помощью построенных описаний объектов и классов в [8] предлагается решать следующие задачи.

Задача идентификации. *Проверить, удовлетворяет ли объект ω или его часть описанию класса $A_k(\overline{x})$ и предъявить эту часть объекта.*

Задача классификации. *Найти все такие номера k, что верна формула $A_k(\overline{\omega})$.*

Задача анализа. *Найти и классифицировать все части τ объекта ω, для которых $A_k(\overline{\tau})$.*

Решение задач идентификации, классификации и анализа для распознавания сложного объекта сведено в к доказательству соответственно логических следований[2]

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A_k(\overline{x}), \tag{1}$$

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} A_k(\overline{\omega}), \tag{2}$$

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} \exists \overline{x}_{\neq}\ A_k(\overline{x}). \tag{3}$$

---

[1] Здесь и далее посредством $\overline{x}$ обозначается список элементов конечного множества x, соответствующий некоторой перестановке номеров его элементов. Тот факт, что элементами списка $\overline{x}$ являются элементы множества y, будем записывать в виде $x \subseteq y$.

[2] Для того, чтобы записать, что значения для переменных списка $\overline{x}$, удовлетворяющие формуле $A(\overline{x})$, различны, вместо формулы

$$\exists x_1 ... \exists x_m (\&_{i=1}^{m} \&_{j=i+1}^{m} (x_i \neq x_j)\ \&\ A(x_1, ..., x_m))$$

будет использоваться обозначение

$$\exists \overline{x}_{\neq} A(\overline{x}).$$

Строго говоря, вместо формул (1), (2), (3) следовало бы писать соответственно

$$S(\omega) \Rightarrow (?\overline{x}_{\neq})\ A_k(\overline{x}), \qquad (1')$$

$$S(\omega) \Rightarrow (?k_{k=1}^{M})A_k(\overline{\omega}), \qquad (2')$$

$$S(\omega) \Rightarrow (?k_{k=1}^{M})(?\overline{x}_{\neq})\ A_k(\overline{x}), \qquad (3')$$

но рассматриваемые алгоритмы доказательства логических следований не только отвечают на вопрос *«существвует ли ... ?»*, но и предъявляют значения для переменных [2].

Заметим, что для того, чтобы уметь доказывать (1), (2), (3), достаточно уметь доказывать логическое следование

$$S(\omega) \Rightarrow \exists\overline{x}_{\neq}\ A(\overline{x}), \qquad (4)$$

где $A(\overline{x})$ — элементарная конъюнкция атомарных формул и их отрицаний. В [4,8] доказаны оценки числа шагов алгоритмов, решающих задачу (4), а также задачи (1), (2), (3). Эти оценки имеют экспоненциальный от длины записи формулы $A(\overline{x})$ вид. Для алгоритма полного перебора в показателе оценки находится количество переменных формулы $A(\overline{x})$, а для алгоритмов, основанных на построении вывода в исчислении предикатов, в показателе оценки находится количество атомарных формул, входящих в формулу $A(\overline{x})$.

Доказана NP-полнота задач (1), (2), (3) и, следовательно, NP-трудность задач (1'), (2'), (3').

3. Многоуровневое описание классов. Для уменьшения числа шагов работы алгоритмов, решающих описанные задачи, в [6] предложено многоуровневое описание классов распознаваемых объектов, по сути своей являющееся иерархическим описанием классов и учитывающее составляющие конструкции объектов. В [5] описана возможность построения логико-предикатной нейронной сети на основании уже имеющегося многуровневого описания классов.

Алгоритм автоматического построения многуровневого описания класса, позволяющий выделить обобщённые характеристики объектов, присущие объектам одного класса, описан в [9]. Этот алгоритм базируется на понятии неполной выводимости предикатной формулы, описанном в [7].

Рассматриваются объекты, структура которых позволяет выделить достаточно простые их части и дать описание объекта в терминах свойств этих частей и отношений между ними. В частности, это можно сделать, выделяя «часто» встречающиеся подформулы $P_i^1(\overline{y}_i^1)$, формул $A_k(\overline{x})$ «небольшой сложности». При этом записывается система равносильностей вида ${p_i}^1(y^1) \Leftrightarrow P_i^1(\overline{y}_i^1)$, где ${p_i}^1$ – новые предикаты, которые будем называть предикатами 1-го уровня, а переменные y_i^1 – новые переменные для списков исходных переменных, которые будем называть переменными 1-го уровня.

Обозначим формулы, полученные из $A_k(\overline{x}_k)$ путем замены всех вхождений формул вида $P_i^1(\overline{y}_i^1)$ на атомарные формулы $p_i^1(x_i^1)$ (при $y_i^1 \subseteq x$) посредством $A_k^1(\overline{x}_k^1)$. Здесь $\overline{x}_k^1$ – список всех переменных формулы $A_k^1(\overline{x}^1)$, состоящий как из некоторых (быть может всех) исходных переменных формулы $A_k(\overline{x}_k)$, так и из переменных первого уровня, появившихся в формуле $A_k^1(\overline{x}_k^1)$. Такие формулы $A_k^1(\overline{x}_k^1)$ можно рассматривать как описания классов в терминах предикатов исходного (нулевого) и первого уровней.

Описанием объекта $S^1(\omega)$ первого уровня назовем множество всех атомарных формул вида $p_i^1(\omega_{ij}^1)$, для которых истинна определяющая подформула $P_i^1(\overline{\tau}_{ij}^1)$ при $\tau_{ij}^1 \subset \omega$, а объект первого уровня ω_{ij}^1 представляет из себя список исходных объектов $\overline{\tau}_{ij}^1$.

Процедуру выделения «часто» встречающихся подформул «небольшой сложности» можно повторить с формулами $A_k^1(\overline{x}^1)$.

В результате построения составных предикатов (т.е. предикатов различных уровней) и многоуровневого описания классов исходное множество описаний классов $\{A_k(\overline{x})\}$ может быть записано с помощью равносильной ей многоуровневой системы описаний классов вида

$$\begin{cases} & A_k^L(\overline{x}^L) & \\ p_1^1(x_1^1) & \Leftrightarrow & P_1^1(\overline{y}_1^1) \\ & \vdots & \\ p_{n_1}^1(x_{n_1}^1) & \Leftrightarrow & P_{n_1}^1(\overline{y}_{n_1}^1) \\ & \vdots & \\ p_i^l(x_i^l) & \Leftrightarrow & P_i^l(\overline{y}_i^l) \\ & \vdots & \\ p_{n_L}^L(x_{n_L}^L) & \Leftrightarrow & P_{n_L}^L(\overline{y}_{n_L}^L) \end{cases}.$$

Алгоритм многоуровневого распознавания.

Проверка следования (4) при использовании L-уровневого описания разбивается на последовательное в цикле при $l = 1, ..., L$ выполнение п.п. 1 – 4 с последующим выполнением п. 5.

1. Проверка следований $S^{l-1}(\omega) \Rightarrow \exists \overline{y}^l_{i\neq} P^l_i(\overline{y}^l_i)$ $(i = 1, ..., n^l_i)$ с нахождением тех наборов $\overline{\tau}^l_i$ значений исходных констант для списка переменных $\overline{y}^l_i$, при которых $S^{l-1}(\omega) \Rightarrow P^l_i(\overline{\tau}^l_i)$ $(i = 1, ..., n^l_i)$.

2. Введение новых атомарных формул $p^l_i(y^l_i)$ l-го уровня, определяемых равносильностями $p^l_i(y^l_i) \Leftrightarrow P^l_i(\overline{y}^l_i)$ с новыми переменными y^l_i l-го уровня для списков переменных $\overline{y}^l_i$.

3. Замена в формуле $A^{l-1}(\overline{x}^{l-1})$ всех подформул вида $P^l_i(\overline{y}^l_i)$ $(i = 1, ..., n^l_i)$ на атомарные формулы $p^l_i(y^l_i)$ и получение формулы $A^l(\overline{x}^l)$.

4. Добавление в $S^{l-1}(\omega)$ постоянных атомарных формул l-го уровня вида $p^l_i(\tau^l_i)$, где τ^l_i — новые константы, задающие списки констант $\overline{\tau}^l_i$ $(i = 1, ..., n^l_i)$ и получение описания объекта l-го уровня $S^l(\omega)$.

5. Проверка следования $S^L(\omega) \Rightarrow \exists \overline{x}^L{}_{\neq} A^L(\overline{x}^L)$ с нахождением тех наборов $\overline{\tau}^L$ значений исходных констант для списка переменных $\overline{x}^L$, при которых $S^L(\omega) \Rightarrow A^L(\overline{\tau}^L)$.

В [8] доказаны оценки изменения числа шагов проверки (4) при использовании двухуровневого описания классов и приведены модельные примеры, иллюстрирующие существенное уменьшение показателя экспоненты при его использовании. Однако там применяется эвристическое выделение общих подформул для построения двухуровневого описания классов. В [9] описан алгоритм построения многоуровневого описания класса по обучающей выборке, в основе которого лежит понятие неполной выводимости.

4. Понятие неполной выводимости формулы. Понятие неполной выводимости предикатной формулы было введено в [7] для распознавания объектов с неполной информацией.

Рассматривается задача проверки того, что из истинности всех формул множества $S(\omega)$ следует истинность $A(\overline{x})$ или некоторой её максимальной подформулы $\widetilde{A}(\overline{y})$ на наборе различных констант из ω, где список переменных $\overline{y}$ является подсписком списка переменных $\overline{x}$.

Пусть a и $\widetilde{a}$ – количества атомарных формул в элементарных конъюнкциях $A(\overline{x})$ и в $\widetilde{A}(\overline{y})$ соответственно, m и $\widetilde{m}$ – количества предметных переменных в $A(\overline{x})$ и $\widetilde{A}(\overline{y})$ соответственно.

Числа q и r вычисляются по формулам $q = \frac{\widetilde{a}}{a}$, $r = \frac{\widetilde{m}}{m}$ и характеризуют степень совпадения формул $A(\overline{x})$ и $\widetilde{A}(\overline{y})$. В этом случае подформула $\widetilde{A}(\overline{y})$ называется (q, r)-фрагментом формулы $A(\overline{x})$.

Подформула $\widetilde{A}(\overline{y})$ называется максимальной подформулой элементарной конъюнкции $A(\overline{x})$, если она является её (q, r)-фрагментом с максимальным среди всех (q, r)-фрагментов значением параметра q. То есть для $\widetilde{A}(\overline{y})$ справедливо $S(\omega) \Rightarrow \exists \overline{y}_{\neq} \widetilde{A}(\overline{y})$ и ни для какой подформулы формулы $A(\overline{x})$, с большим значением параметра q, это следствие не выполняется.

Возможно другое определение значений параметров q и r. Пусть предикатным символам, задающим признаки объектов, приписаны веса w_i $(i = 1, \ldots, n)$, а предметным переменным, входящим в формулы, приписаны соответственно веса v_j $(j = 1, \ldots, m)$ и $\widetilde{v}_j$ $(j = 1, \ldots, m')$. Тогда $q_w = \frac{\widetilde{w}}{w}$, $r_v = \frac{\widetilde{v}}{v}$, где w и $\widetilde{w}$ – суммы весов предикатных формул в $A(\overline{x})$ и в $\widetilde{A}(\overline{y})$ соответственно, v и $\widetilde{v}$ – суммы весов предметных переменных в $A(\overline{x})$ и $\widetilde{A}(\overline{y})$ соответственно.

Параметр q, так же как и параметр q_w, характеризует, насколько информативен фрагмент, содержащий лишь r-ую (r_v-ую) часть переменных.

Задача нахождения максимального (q, r)-фрагмента формулы $\widetilde{A}(\overline{y})$ при условии справедливости множества постоянных атомарных формул $S(\omega)$ называется задачей проверки неполной выводимости этой формулы. В [7] приведён один из возможных алгоритмов её решения.

5. Нахождение наибольшей общей подформулы двух формул. Понятие неполной выводимости из множества постоянных атомарных формул или их отрицаний легко обобщается до понятия неполной выводимости двух элементарных конъюнкций.

Пусть $A(\overline{x})$ и $B(\overline{y})$ – две элементарные конъюнкции предикатных формул со списками предметных переменных $\overline{x}$ и $\overline{y}$ соответственно. Проверка неполной выводимости $A(\overline{x}) \Rightarrow_P \exists \overline{y}_{\neq} B(\overline{y})$ заключается в нахождении такого максимального (q, r)-фрагмента $Q_{AB}(\overline{z})$ формулы $B(\overline{y})$ и такой подстановки $\lambda_{AQ} = |^{z}_{y'}$ списка переменных $\overline{y'}$ из $\overline{y}$ вместо переменных списка $\overline{z}$, что $Q_{AB}(\overline{y'})$ является максимальной подформулой формулы $B(\overline{y})$, такой что $A(\overline{x}) \Rightarrow \exists \overline{y'}_{\neq} Q_{AB}(\overline{y'})$.

Аналогично при проверке неполной выводимости $B(\overline{y}) \Rightarrow_P \exists \overline{x}_{\neq} A(\overline{x})$ получаем максимальный (q, r)-фрагмент $Q_{BA}(\overline{z})$ формулы $A(\overline{x})$ и такую подстановку $\lambda_{BQ} = |^{z}_{x'}$ списка переменных $\overline{x'}$ из $\overline{x}$ вместо переменных списка $\overline{z}$, что $Q_{BA}(\overline{y'})$ является максимальной подформулой формулы $A(\overline{x})$, такой что $B(\overline{y}) \Rightarrow \exists \overline{x'}_{\neq} Q_{BA}(\overline{x'})$.

Можно доказать, что формулы $Q_{AB}(\overline{y'})$ и $Q_{BA}(\overline{x'})$ совпадают с точностью до имён переменных. В качестве максимальной (с точностью до имён переменных) общей подформулы двух элементарных конъюнкций $A(\overline{x})$ и $B(\overline{y})$ можно взять любую из них. Обозначим такую подформулу посредством $Q(\overline{z})$.

Найденные в процессе проверки неполной выводимости подстановки λ_{AQ} и λ_{BQ} обеспечивают возможность такого переименования переменных из $\overline{z}$, что $Q(\overline{z})$ становится в точности общей подформулой элементарных конъюнкций $A(\overline{x})$ и $B(\overline{y})$ соответственно. Эти подстановки назовём унификаторами формулы $Q(\overline{z})$ с элементарными конъюнкциями $A(\overline{x})$ и $B(\overline{y})$.

6. Построение многоуровневого описания классов. Понятие неполной выводимости формулы позволяет разработать подход к выделению подформул с требуемыми свойствами [9].

Алгоритм построения многоуровневого описания.

1. Для каждой пары элементарных конъюнкций $A_i(\overline{x}_i)$ и $A_j(\overline{x}_j)$, входящих в описания классов, посредством проверки неполной выводимости для $A_i(\overline{x}_i) \Rightarrow_P \exists \overline{x}_{j\neq}\, A_j(\overline{x}_j)$ выделяем их максимальную (с точностью до имён предметных переменных) подформулу $Q^1_{ij}(\overline{x}_{ij})$.

При использовании как алгоритма полного перебора, так и алгоритма, основанного на построении вывода в исчислении предикатов, найденная формула $Q^1_{ij}(\overline{x}_{ij})$ является в точности подформулой элементарной конъюнкции $A_j(\overline{x}_j)$, поэтому унификатор λ_{iQ} является тождественной подстановкой. При этом будет найден унификатор λ_{jQ}.

2. Повторяем процесс выделения общих подформул для $Q^{l-1}_{i_1\ldots i_{2^{l-1}}}(\overline{x}_{i_1\ldots i_{2^{l-1}}})$ и $Q^{l-1}_{j_1\ldots j_{2^{l-1}}}(\overline{x}_{j_1\ldots j_{2^{l-1}}})$, получив их общие (с точностью до имён предметных переменных) подформулы $Q^{l}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}}(\overline{x}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}})$ $(l = 2, \ldots, L)$ и унификаторы для соответствующих подформул. Процесс завершится, так как на каждой итерации длины подформул уменьшаются.

3. Выберем среди подформул $Q^{l}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}}(\overline{x}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}})$ минимальные (по числу переменных для применения алгоритма полного перебора или по числу атомарных формул для применения алгоритмов, основанных на построении вывода в исчислении предикатов) и обозначим их посредством $P^1_i(\overline{y}^1_i)$ $(i = 1, \ldots, n_1)$.

4. Формулы $P^{l+1}_i(\overline{y}^{l+1}_i)$ $(i = 1, \ldots, n_{l+1},\, l = 2, \ldots, L)$ строятся из выделенных ранее подформул $Q^{l}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}}(\overline{x}_{i_1\ldots i_{2^{l-1}}j_1\ldots j_{2^{l-1}}})$ с учётом минимизации требуемых параметров и того, что подформулы вида $P^1_i(\overline{y}^1_i)$ в них заменены на новые атомарные формулы $p^1_i(y^1_i)$, определяемые равносильностями $p^1_i(y^1_i) \Leftrightarrow P^1_i(\overline{y}^1_i)$.

7. Формирование логико-предикатной сети. На стадии обучения для формирования обучающего блока сети предлагается обучающая выборка, содержащая описания объектов с указанием классов, которым они принадлежат. В описании каждого объекта различные константы заменяются различными переменными и между атомарными формулами ставится знак &.

Описанием класса служит дизъюнкция так полученных элементарных конъюнкций. Затем используется *Алгоритм построения многоуровневого описания* из конъюнкций, соответствующих объектам обучающей выборки.

Полученные в обучающем блоке формулы $P_i^1(\overline{y}_i^1)$ $(i = 1, \dots n_l,$ $l = 1, \dots, L)$ служат содержимым ячеек l-го уровня решающего блока. Последний уровень составляют формулы $A_k^L(\overline{x_k^L})$.

В процессе распознавания используется только решающий блок. Он работает в соответствии с *Алгоритмом многоуровневого распознавания*. Если в процессе использования построенной сети обнаруживается неправильное распознавание объекта, то возможно дообучение сети посредством добавления описания неправильно классифицированного объекта к первому слою обучающего блока и выделению общих подформул этого описания из уже имеющихся. После этого происходит перестройка решающего блока. Схема логико-предикатной сети представлена на рисунке 1.

8. Модельный пример формирования сети. Пример контурных изображений взят из [1].

Пусть дана обучающая выборка класса контурных изображений «ящиков», представленная на рисунке 2.

Каждый объект – это совокупность вершин в контурном изображении. Описания объектов и классов объектов заданы в терминах исходных предикатов V и L, определённых на рисунке 3 и задающих отношения между вершинами.

Описание класса «ящиков» содержит 4 элементарные конъюнкции, каждая из которых состоит из всех атомарных формул, истинных для соответствующего изображения, в которых имя вершины i заменено на переменную x_i. Так, например, изображение b на рисунке 2 описывает элементарная конъюнкция $A_2(x_1, \dots, x_8) =$

$$V(x_1, x_4, x_2) \& V(x_2, x_1, x_6) \& V(x_2, x_6, x_3) \& V(x_2, x_1, x_3) \&$$

$$V(x_3, x_2, x_8) \& V(x_4, x_5, x_1) \& V(x_4, x_6, x_1) \& V(x_4, x_7, x_5) \&$$

$$V(x_4,x_7,x_6)\&V(x_4,x_7,x_1)\&V(x_5,x_4,x_7)\&V(x_5,x_7,x_6)\&$$
$$V(x_6,x_2,x_5)\&V(x_6,x_2,x_4)\&V(x_6,x_5,x_8)\&V(x_6,x_4,x_8)\&$$
$$V(x_6,x_8,x_2)\&V(x_7,x_5,x_4)\&V(x_7,x_8,x_5)\&V(x_7,x_8,x_4\&$$
$$V(x_8,x_3,x_6)\&V(x_8,x_6,x_7)\&V(x_8,x_3,x_7)\&L(x_5,x_4,x_6).$$

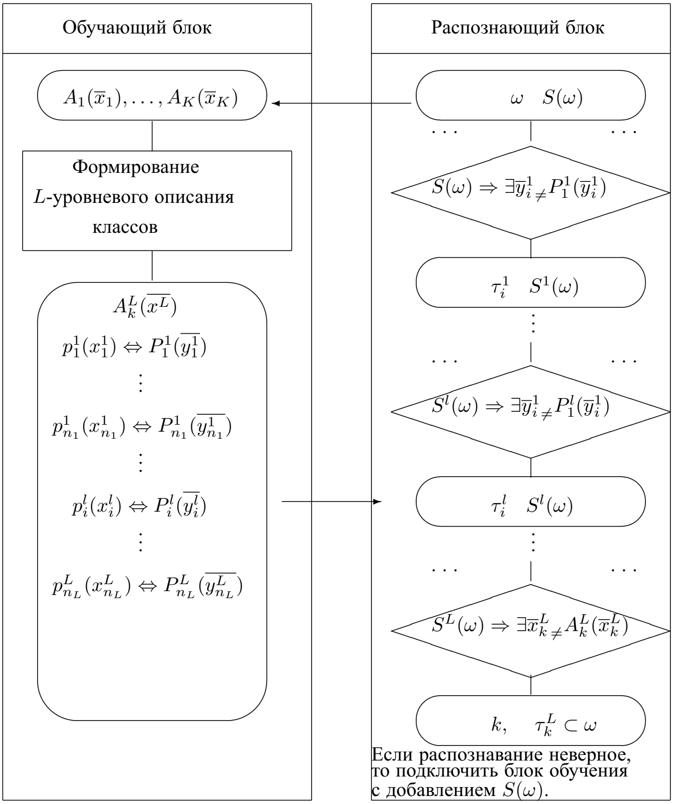


Рис. 1. Схема логико-предикатной сети

Попарная проверка частичной выводимости между этими элементарными конъюнкциями позволяет выделить их максимальные общие подформулы, соответствующие изображениям на рисунке 4.

В процессе попарного выделения общих подформул формул $A_i(\overline{x}_i)$ и $A_j(\overline{x}_j)$ $(i = 1, ..., 3,\ j = i + 1, \ldots, 4)$ получили их максимальные (с точностью до имён переменных) подформулы, соответствующие

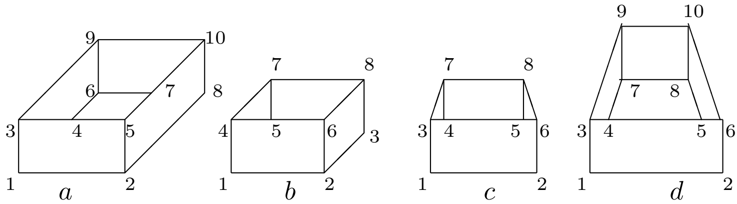
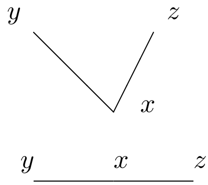


Рис. 2. Обучающая выборка



$$V(x,y,z) \Longleftrightarrow (\angle yxz < \pi)$$

$L(x,y,z,) \Longleftrightarrow x$ лежит на отрезке с концами y и z

Рис. 3. Исходные предикаты

изображениям, представленным на рисунке 4. Так, например, подформула $Q^1_{2,4}(x_2,x_3,x_4,x_5,x_9,x_{10}) =$

$$V(x_1,x_3,x_2)\&V(x_2,x_1,x_5)\&V(x_3,x_4,x_1)\&V(x_3,x_5,x_1)\&$$

$$V(x_3,x_9,x_4)\&V(x_3,x_9,x_5)\&V(x_3,x_9,x_1)\&V(x_5,x_2,x_4)\&$$

$$V(x_5,x_2,x_3)\&V(x_9,x_{10},x_4)\&V(x_9,x_4,x_3)\&L(x_4,x_3,x_5)$$

описывает изображение bd на рисунке 4.

После попарного выделения максимальных общих подформул формул $Q^1_{i_1,i_2}(x^1_{i_1,i_2})$ и $Q^1_{j_1,j_2}(x^1_{j_1,j_2})$ получена единственная их максимальная общая подформула $Q^2(x_1,x_2,x_3,x_4,x_5,x_9,x_{10}) =$

$$V(x_1,x_3,x_2)\&V(x_2,x_1,x_5)\&V(x_3,x_4,x_1)\&V(x_3,x_5,x_1)\&$$

$$V(x_3,x_9,x_4)\&V(x_3,x_9,x_5)\&V(x_3,x_9,x_1)\&V(x_5,x_2,x_4)\&$$

$$V(x_5,x_2,x_3)\&V(x_9,x_{10},x_3)\&L(x_4,x_3,x_5),$$

соответствующая изображению ad на рисунке 4.

Элементарная конъюнкция $P^1(x_1,x_2,x_3,x_4,x_5,x_9,x_{10}) = V(x_1,x_3,x_2)\&V(x_2,x_1,x_5)\&V(x_3,x_4,x_1)\&V(x_3,x_5,x_1)\&$

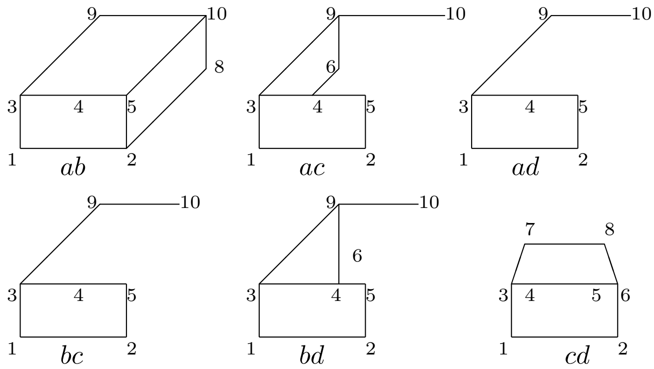


Рис. 4. Изображения, соответствующие выделенным подформулам

$V(x_3, x_9, x_4)\&V(x_3, x_9, x_5)\&V(x_3, x_9, x_1)\&V(x_5, x_2, x_4)\&$ $V(x_5, x_2, x_3)\&V(x_9, x_{10}, x_3)\&L(x_4, x_3, x_5)$, соответствующая этому изображению, определяет предикат 1-го уровня $p^1(x^1)$. Переменная первого уровня x^1 – это переменная для списка из 7 исходных переменных.

При этом, учитывая унификаторы, которые были определены при выделении всех общих (с точностью до имён переменных) подформул, при подстановке в формулы $A_i(\overline{x}_i)$ $(i = 1, ..., 4)$ значением переменной x^1 будут соответственно списки $(x_1, x_2, x_3, x_4, x_5, x_9, x_{10})$, $(x_1, x_2, x_4, x_5, x_6, x_7, x_8)$, $(x_1, x_2, x_3, x_4, x_6, x_7, x_8)$, $(x_1, x_2, x_3, x_4, x_6, x_9, x_{10})$.

После замены в формулах $A_i(\overline{x}_i)$ $(i = 1, ..., 4)$ подформул вида $P^1(\overline{y}^1)$ $(i = 1, ..., 4)$ с учётом полученных унификаторов на атомарную формулу $p^1(x^1)$ получим двухуровневое описание классов. Например, элементарная конъюнкция для изображения b на рисунке 2 примет вид $A_2^1(x^1, x_1, \ldots, x_8) =$

$$p^1(x^1)\&V(x_2, x_6, x_3)\&V(x_2, x_1, x_3)\&V(x_3, x_2, x_8)\&$$

$$V(x_5, x_4, x_7)\&V(x5, x7, x6)\&V(x6, x5, x8)\&V(x6, x4, x8)\&$$

$$V(x_6, x_8, x_2)\&V(x_7, x_5, x_4)\&V(x_7, x_8, x_5)\&V(x_7, x_8, x_4)\&$$

$$V(x_8, x_3, x_6)\&V(x_8, x_6, x_7)\&V(x_8, x_3, x_7).$$

Следует отметить, что несмотря на то, что формально количество переменных в формуле увеличилось, но при последующем присвоении только переменной x^1 какого-то набора значений констант $(a_1, a_2, a_4, a_5, a_6, a_7, a_8)$, в формуле с неприсвоенными значениями останется только переменная x_3, так как остальным переменным будут присвоены значения, взятые из списка значений для переменной x^1.

Четыре признака второго уровня определяются элементарными конъюнкциями, соответствующими изображениям ab, ac, bd и cd на рисунке 4. При этом эти элементарные конъюнкции содержат признаки 1-го уровня и переменную 1-го уровня.

Элементарные конъюнкции $P_1^2(\overline{y}_1^2)$, $P_2^2(\overline{y}_2^2)$, $P_3^2(\overline{y}_3^2)$, $P_4^2(\overline{y}_4^2)$, соответствующие изображениям ab, ac, bd, cd на рисунке 4 и записанные с использованием предиката $p^1(x^1)$ определяют предикаты второго уровня $p_1^2(x_1^2)$, $p_2^2(x_2^2)$, $p_3^2(x_3^2)$, $p_4^2(x_4^2)$.

Например, подформула $P_1^2(\overline{y}_1^2) =$

$$p^1(x^1)\&V(x_2, x_5, x8)\&V(x_2, x_1, x_8)\&V(x_5, x_4, x_{10})\&V(x_5, x_3, x_{10})\&$$

$$V(x_8, x_2, x_{10})\&V(x_{10}, x_8, x_5)\&V(x_{10}, x_5, x_9)\&V(x_{10}, x_8, x_9)$$

соответствует изображению ab на рисунке 4. Здесь y_1^2 – переменная для списка, состоящего из переменных $(x^1, x_1, x_2, x_4, x_5, x_8, x_9, x_{10})$, причём переменная 1-го уровня x^1 – это переменная для списка исходных переменных $(x_1, x_2, x_3, x_4, x_5, x_9, x_{10})$. Заметим, что при определении значения для переменной 1-го уровня x^1 только переменная x_8 не получила значения.

После замены в формулах $A_i^1(\overline{x}_i^1)$ подформул $P_j^2(\overline{y}_i^2)$ $(i, j = 1, ..., 4)$ на атомарные формулы $p_j^2(x_j^2)$ получим двухуровневое описание классов. Например, элементарная конъюнкция для изображения a на рисунке 2 примет вид $A_1^2(x_1^2, x_3, x_4, x_5, x_6, x_7, x_9, x_{10},) =$

$$p^1(x_1^2)\&V(x_4, x_3, x_6)\&V(x_4, x_6, x_5)\&V(x_6, x_4, x_9)\&$$

$$V(x_6, x_9, x_7)\&V(x_9, x_7, x_4)\&V(x_7, x_5, x_6)\&V(x_7, x_6, x_{10})\&$$

$$V(x_9, x_6, x_3)\&V(x_9, x_{10}, x_6)\&V(x_9, x_{10}, x_3)\&L(x_7, x_5, x_{10}).$$

Следует отметить, что все исходные переменные, кроме переменной x_6, входят в список переменных, определяющих x^2 .

Работа обучающего блока закончена. Сформирована 3-уровневая сеть. Схематически последовательность проверки 3-уровневой сетью истинности подформул, соответствующих фрагментам изображения, представлена на рисунке 5.

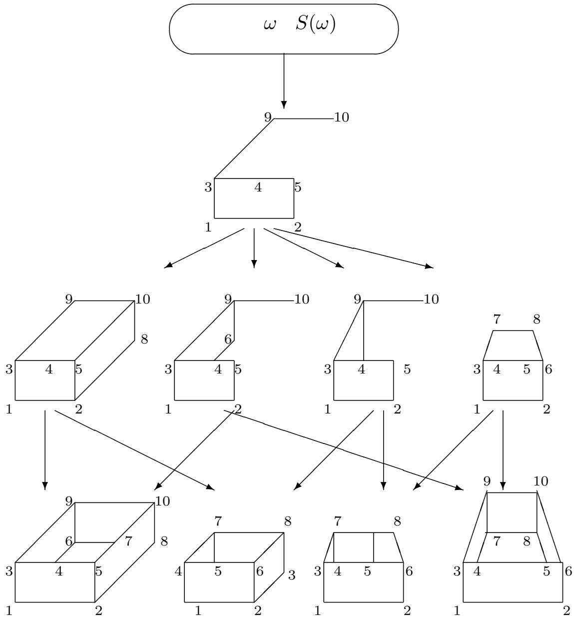


Рис. 5. Фрагменты изображений, выделяемые 3-уровневой сетью

В процессе распознавания все объекты, соответствующие изображениям на рисунке 2, будут распознаны правильно.

Пусть для распознавания представлен объект, изображённый на рисунке 6. Построенная сеть не сможет его распознать, так как формула, определяющая предикат 1-го уровня, не будет истинной на его описании.

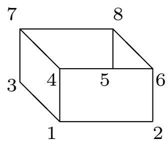


Рис. 6. Контрольное изображение

Добавим элементарную конъюнкцию, соответствующую его описанию, к ранее заданным исходным данным обучающего блока. Попарная проверка неполной выводимости этой элементарной конъюнкции с уже имеющимися в обучающем блоке выделит новые подформулы. При этом предикат 1-го уровня будет задаваться элементарной конъюнкцией, соответствующей изображению на рисунке 7.

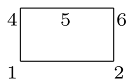


Рис. 7. Изображение, соответствующее новому предикату 1-го уровня

Предикаты 2-го уровня задаются формулами, соответствующими изображениям на рисунке 8.

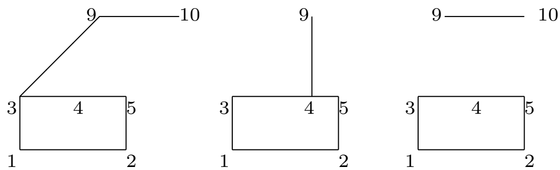


Рис. 8. Изображения, соответствующие новым предикатам 2-го уровня

Множество формул, определяющих предикаты 3-го уровня, совпадает с ранее построенным множеством предикатов 2-го уровня.

Таким образом, распознающий блок перестроен и представляет собой 4-уровневое описание класса. Схематически последовательность проверки 4-уровневой сетью истинности подформул, соответствующих фрагментам изображения, представлена на рисунке 9.

9. Заключение. В статье описан подход к формированию самоперестраивающейся нейронной сети с элементами, реализующими вычисление значения элементарной конъюнкции предикатных формул.

Основной проблемой при конструировании таких сетей в настоящий момент является детальная проработка способа хранения и передачи унификаторов, позволяющих отождествлять переменные и константы в выделенных подформулах и в формулах, в которые они подставляются.

Непроработанным также остаётся вопрос, какие же из выделенных подформул подставлять в исходные, если возможно несколько вариантов подстановок.

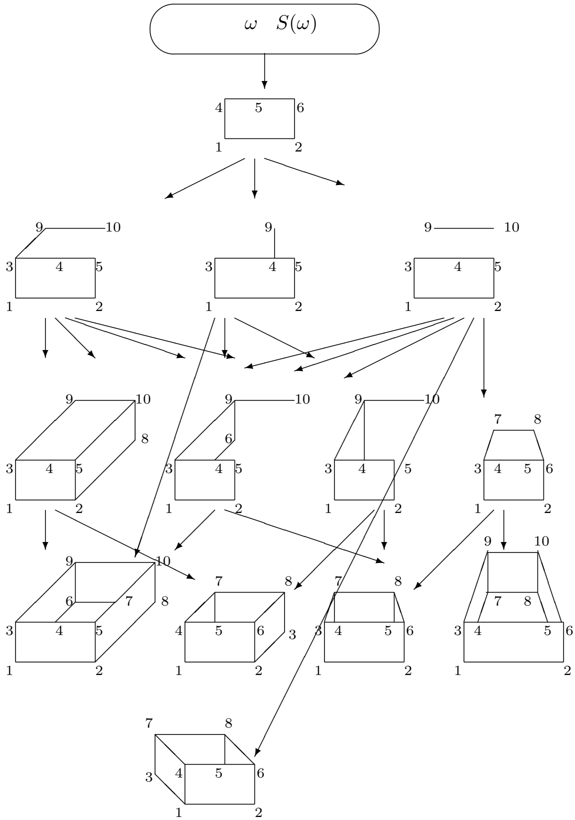


Рис. 9. Фрагменты изображений, выделяемые 4-уровневой сетью

Возможны варианты, когда выделенная подформула содержит несколько переменных более низкого уровня и, следовательно, задаёт отношение между ними. Интересны возможности использования взвешенных предикатов и переменных. По-видимому, эти веса должны меняться в зависимости от «верного» или «ошибочного» распознавания.

Список литературы

1. *Дуда Р., Харт П.* Распознавание образов и анализ сцен // М.: Мир, 1976. 511 с.
2. *Клини С.* Математическая логика // М.: Мир, 1973. 480 с.

3. *Комашинский В.И., Смирнов Д.А.* Нейронные сети и их применение в системах управления и связи // М.: Горячая линия–Телеком, 2002. 94 с.
4. *Косовская Т.М.* Доказательства оценок числа шагов решения некоторых задач распознавания образов, имеющих логические описания // Вестник Санкт-Петербургского университета. Сер. 1. 2007. Вып. 4. С. 82–90.
5. *Косовская Т.М., Тимофеев А.В.* Иерархическое описание классов и нейросетевое распознавание сложных образов // Нейрокомпьютеры: разработка, применение. 2007. № 6. С. 30 – 33.
6. *Косовская Т.М.* Многоуровневые описания классов для уменьшения числа шагов решения задач распознавания образов, описываемых формулами исчисления предикатов // Вестн. С.-Петербург.ун-та. Сер. 10. 2008. Вып. 1. С. 64 – 72.
7. *Косовская Т. М.* Частичная выводимость предикатных формул как средство распознавания объектов с неполной информацией // Вестн. С.-Петербург.ун-та. Сер. 10. 2009. Вып. 1. С. 74 – 84.
8. *Косовская Т.М.* Некоторые задачи искусственного интеллекта, допускающие формализацию на языке исчисления предикатов, и оценки числа шагов их решения // Труды СПИИРАН. 2010. Вып. 14. С. 58 – 75.
9. *Косовская Т.М.* Подход к решению задачи построения многоуровневого описания классов на языке исчисления предикатов // Труды СПИИРАН. 2014. № 3(34). С. 204 – 217.
10. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход, 2-е изд. // Пер. с англ. М.: Издательский дом “Вильямс”, 2006. 1408 с.

References

1. Duda R.O., Hart P.E. Pattern Classification and Scene Analysis. A Wiley-Interscience Publication, John Wiley & Sons, New York London Sydney Toronto, 1973. 482 p. (Russ. ed.: Duda R., Hart P. Raspoznavanie obrazov i analiz scen. M.: Mir, 1976. 511p p.).
2. Kleene S.C. Mathematical logic. Dover Publications, New York: Wiley, 1967. (Russ. ed.: Klini S. Matematicheskaja logika. M.: Mir, 1973. 480 p.).
3. Komashinskiy V.I., Smirnov D.A. *Nejronnye seti i ih primenenie v sistemah upravleniya i svyazi* [Neural networks and their application in control systems and communication]. Moscow: Goryachaya liniya - Telekom, 2002. (In Russ.).
4. Kosovskaya T.M. [Proofs of the number of steps bounds for solving of some pattern recognition problems with logical description]. *Vestnik Sankt-Peterburgskogo Universiteta – Bulletin of St. Petersburg State University*. 2007, No. 4, pp. 82–90. (In Russ.).
5. Kosovskaya T.M., Timofeev A.V. [Hierarhical description of classes and neuron network recognition of complex patterns]. *Nejrokompjutery:*

razrabotka i primenenie – Neurocomputers: development, application. 2007. No. 6. pp. 30–33. (In Russ.)
6. Kosovskaya T.M. [Level descriptions of classes for decreasing step number of pattern recognition problem solving described by predicate calculus formulas]. *Vestnik Sankt-Peterburgskogo Universiteta – Bulletin of St. Petersburg State University*. 2008, vol. 10. No. 1, pp. 64–72. (In Russ.).
7. Kosovskaya T.M. [Partial hatchability predicate formulas as a means of recognition of objects with incomplete information]. *Vestnik Sankt-Peterburgskogo Universiteta – Bulletin of St. Petersburg State University*. 2009, vol. 10. No. 1, pp. 74–84. (In Russ.).
8. Kosovskaya T.M. [Some artificial intelligence problems permitting formalization by means of predicate calculus language and upper bounds of their solution steps]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2010. vol. 14, pp. 58 - 75. (In Russ.).
9. Kosovskaya T.M. [An approach to the construction of a level description of classes by means of a predicate calculus language]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2014. vol. 3(34), pp. 58–75. (In Russ.).
10. Russel S.J., Norvig P. Artificial Intelligence. A Modern Approach. Pearson Education. Inc. 2003. (Russ. ed.: Rassel S., Norvig P. Iskusstvennyj intellekt: sovremennyj podhod, 2-e izd. Per. s angl. M.: Izdatel'skij dom "Vil'jams", 2006. 1408 p.).

Косовская Татьяна Матвеевна — д-р физ.-мат. наук, доцент, профессор математико-механического факультета, Санкт-Петербургский государственный университет (СПбГУ), старший научный сотрудник, Федеральное государственное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: логический подход к решению задач искусственного интеллекта. Число научных публикаций — 85. kosovtm@gmail.com; 14-я линия В.О., д. 39, Санкт-Петербург, 199178; р.т.: +79213232307.

Kosovskaya Tatiana Matveevna — Ph.D., Dr. Sci., associate professor, professor of computer science department, St.Petersburg State University (SPbSU), senior researcher of autonomous robotic systems laboratory, St. Petersburg Institute for Informatics and Automation of Russian Academy of Scientists (SPIIRAS). Research interests: logical approach to the solving of artificial intelligence problems, theory of complexity of algorithms. The number of publications — 85. kosovtm@gmail.com; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +79213232307.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (проект № 14-08-01276-а).

Acknowledgements. This research is supported by RFBR (grant 14-08-01276-a).

РЕФЕРАТ

Косовская Т.М. **Самообучающаяся сеть с ячейками, реализующими предикатные формулы.**

Рассматривается модель перенастраиваемой сети с ячейками, реализующими предикатные формулы, имеющие вид элементарных конъюнкций. В отличие от классических нейронных сетей предлагаемая модель имеет два блока: блок обучения и блок решения.

При ошибках, возникающих при использовании блока решения, подключается блок обучения, который определяет дополнительные составные характеристики, присущие как ранее представленным объектам, так и вновь поступившему, для которого произошла ошибка. Кроме того, конфигурация сети не фиксируется заранее, а меняется каждый раз после работы блока обучения.

Рассматриваются задачи искусственного интеллекта, в которых исследуемый объект представлен как множество элементов, характеризующихся своими свойствами и отношениями между ними. Исследуемый объект задаётся своим описанием, представленным как множество постоянных атомарных формул (или их отрицаний), истинных для этого объекта. Целевые формулы записываются в виде дизъюнкции элементарных конъюнкций атомарных формул.

Поскольку задачи, допускающие такую формализацию, NP-трудны, то встаёт вопрос об уменьшении показателя степени экспоненты, ограничивающей число шагов решения задач. Построение уровневого описания целевых формул посредством выделения из них общих подформул позволяет существенно уменьшить число шагов алгоритмов, решающих рассматриваемые задачи. Такие уровневые описания соответствуют декомпозиции задачи большой размерности на несколько последовательно решаемых задач меньшей размерности. При этом в оценке числа шагов алгоритма вместо большого показателя степени экспоненты, соответствующего размерности исходной задачи, получается сумма, каждое слагаемое которой в показателе степени имеет размерность подзадачи.

В статье предлагается создание сети, в которой обучающий блок формирует многоуровневое описание классов. Работа обучающего блока заключается в создании уровневого описания классов на основе обучающей выборки посредством выделения общих составных характеристик всех элементов обучающей выборки. Описан алгоритм выделения таких составных характеристик, основанный на ранее введённом автором понятии неполной выводимости. Работа распознающего блока происходит в соответствии с приведённым в статье алгоритмом. Приведён модельный пример построения 3-уровневой сети и последующей её перестройки в 4-уровневую.

SUMMARY

Kosovskaya T.M. **Self-training Network with the Sells Implementing Predicate Formulas.**

A model of self-modificated predicate network with cells implementing predicate formulas in the form of elementary conjunction is suggested. Unlike a classical neuron network the proposed model has two blocks: a training block and a recognition block.

If a recognition block has a mistake then the control is transmitted to a training block. Always after a training block implementation the configuration of the recognition block is changed. The base of the proposed predicate network is logic-objective approach to AI problems solving and level description of classes.

Artificial Intelligence problems, an investigated object of which is presented as a set of elements characterized by their properties and relations between them, are under consideration. An investigated object is represented by its description as a set of constant atomic formulas (or their negations) which are true for this object. A goal formula is written in the form of disjunction of elementary conjunctions of atomic formulas.

As all these problems are NP-hard a problem of decreasing for the exponent of the number of steps upper bound for the decision algorithm is very important. Construction of a level description for goal formulas, by means of extracting their common sub-formulas, allows essentially to decrease the number of steps for an algorithm solving a problem under consideration. These descriptions correspond to decomposition of a high-dimensional problem up to several less-dimensional sub-problems which are solved sequentially. In this case the upper bound of number of steps of an algorithm is the sum of terms with exponents equal to dimensions of sub-problems instead of one term which exponent equals to the dimension of the initial problem.

A network with the training block creating a level description of classes is suggested in the paper. The training block implementation consists in the construction of a level description on the base of a training set be means of the extraction common complex characteristics of the training set elements. An algorithm of such an extraction based on the proposed earlier by the author notion of partial deducibility is described in the paper.

The recognition block implementation is made according to the algorithm described in the paper. A model example of 3-level network construction and further its reconstruction up to a 4-level network is presented.