

ИЕРАРХИЧЕСКОЕ ЛОГИЧЕСКОЕ ОПИСАНИЕ И НЕЙРОСЕТЕВОЕ РАСПОЗНАВАНИЕ СЛОЖНЫХ ОБРАЗОВ

Татьяна Косовская, Адиль Тимофеев

Аннотация: Авторы предложили ранее иерархический способ задания описаний классов при решении задач распознавания образов. В данной работе предлагается развитие и использование таких иерархических описаний для параллельного представления сложных образов на базе много-ядерных компьютеров или нейронных сетей.

Ключевые слова: сложные образы, логическое описание классов, нейронная сеть.

Введение

В работах [1–4] авторами был предложен логико-аксиоматический подход к решению ряда задач распознавания образов. В работах [1, 2] в рамках этого подхода был предложен иерархический способ задания описаний классов, предназначенный в первую очередь для уменьшения сложности решения рассмотренных задач. В настоящей работе предлагается использовать такие иерархические описания для классов (ОК) для их параллельного представления на базе нейронных сетей. Нейросетевое представление логических описаний классов и решающих правил обеспечивает массовый параллелизм и высокое быстродействие при распознавании сложных образов.

Для построения логических ОК было предложено использовать следующие методы и технологии:

- словесное (лингвистическое) описание классов, взятое из специальной учебной, научной или справочной литературы [2–4];
- непосредственное описание обучающей выборки в терминах заданных предикатов [4];
- логико-частотный метод [3, 4];
- тестовые таблицы [5];
- диофантовые нейронные сети [6–8];
- мульти-агентные технологии принятия решений коллективом нейронных сетей [9].

1. Постановка задач логического распознавания сложных образов

Пусть на частях (фрагментах) x , входящих в объекты произвольной природы ω из множества Ω , задан набор предикатов, характеризующих свойства и отношения между элементами, фрагмента x объекта $\omega \in \Omega$.

Пусть задано разбиение множества Ω на M (непересекающихся или, возможно, пересекающихся) классов (образов) вида

$$\Omega = \bigcup_{k=1}^M \Omega_k.$$

Описанием $S(\omega)$ объекта ω называется набор атомарных формул вида $p_i(x)$ или $\neg p_i(x)$, истинных для x , выписанных для всех возможных частей (фрагментов) x объекта ω .

Описанием класса (ОК) Ω_k называется такая логическая формула $A_k(\omega)$, что

1. $A_k(\omega)$ содержит в качестве атомарных только формулы вида $p_i(x)$, где $x \subseteq \omega$;
2. $A_k(\omega)$ не содержит кванторов;
3. если истинна формула $A_k(\omega)$, то $\omega \in \Omega_k$

ОК всегда может быть записано в виде логической формулы

$$\bigvee_{j=1}^{J_k} \&_{i \in I_k^j} \&_{x \subseteq \omega_{i,j}} p_i^{\alpha_{ijx}}(x), \quad (1)$$

где J_k – натуральное число, $I_k^j \subseteq \{1, \dots, n\}$, $\omega_{i,j} \subseteq \omega$, α_{ijx} – логические константы. Выражение p^α используется как сокращение для записи p или $\neg p$ в зависимости от того, является ли α константой И (1) или Л (0) соответственно.

С помощью построенных логических ОК предлагается решать следующие задачи распознавания простых и сложных образов:

1. Задача идентификации. Определить, принадлежит ли объект ω или его часть x классу Ω или его часть классу Ω_k .

Эта задача в работах [3, 4] сведена к доказательству выводимости формулы

$$\exists y (y \subseteq \omega \& A_k(y)) \quad (2)$$

из описания распознаваемого объекта $S(\omega)$.

2. Задача классификации. Найти все такие номера классов k , что $\omega \in \Omega_k$.

Эта задача в работах [3, 4] сведена к доказательству выводимости формулы

$$\bigvee_{j=1}^M A_k(\omega) \quad (3)$$

из описания распознаваемого объекта $S(\omega)$ с указанием всех таких номеров k , для которых соответствующий дизъюнктивный член истинен на ω .

3. Задача анализа сложного объекта. Найти и классифицировать все части x объекта ω , для которых $x \in \Omega$.

Эта задача в работах [3, 4] сведена к доказательству выводимости формулы

$$\bigvee_{j=1}^M \exists x (x \subseteq \omega \& A_k(y)) \quad (4)$$

из описания распознаваемого сложного объекта $S(\omega)$ с указанием (локализацией) и идентификацией всех частей сложного объекта ω , поддающихся классификации, т.е. определить принадлежность к тому или иному классу каждого выделенного фрагмента.

Число шагов работы алгоритма, строящего вывод указанных формул, особенно в случае решения задачи анализа сложного объекта, может быть достаточно велико. Для сокращения числа шагов решения задач распознавания в [2] было предложено более экономное иерархическое описание классов, допускающее распараллеливание при реализации на много-ядерных компьютерах и нейронных сетях.

2. Иерархическое логическое описание классов

Рассматриваются объекты, структура которых позволяет выделить более простые их составляющие, т.е. фрагменты, и дать описание объекта в терминах свойств этих составляющих (фрагментов) и отношений между ними. В частности, это можно сделать, выделяя "часто" встречающиеся подформулы формул $A_k(\omega)$ "небольшой сложности". При этом формируется система эквивалентностей вида

$$p_i^l(x) \leftrightarrow P_i^l(x), \quad (5)$$

где p_i^l – предикаты l -го уровня, P_i^l – подформулы формул $A_k(\omega)$.

Более точно, будем говорить, что система исходных предикатов P_1, \dots, P_n определяет свойства и отношения первого уровня и иногда писать p_i^1 вместо P_i .

Зафиксируем целые положительные числа r и N . Они будут характеризовать такие нечеткие понятия, как "небольшая сложность" подформулы (количество переменных в подформуле меньше r) и "часто" (количество вхождений, с точностью до имен переменных, данной подформулы в уже имеющуюся больше N).

Выделим все часто встречающиеся подформулы небольшой сложности формул $A_k(\omega)$ и обозначим их $P_i^2(x)$ (где x – список переменных, входящих в подформулу).

Обозначим предикаты, задаваемые этими подформулами посредством $p_i^2 (i = 1, \dots, n_2)$ и будем называть их составными предикатами второго уровня. Эти предикаты определяются соотношениями

$$p_i^2(x) \leftrightarrow P_i^2(x) \quad (6)$$

Обозначим формулы, полученные из $A_k(\omega)$ путем замены всех вхождений формул вида $P_i^2(x)$ на атомарные формулы $p_i^2(x)$ (при $x \subseteq \omega$) через $A_k^2(\omega)$. Такие формулы можно рассматривать как логические ОК в терминах предикатов первого и второго уровней.

Процедуру выделения часто встречающихся (типичных) подформул небольшой сложности можно повторить с формулами $A_k^2(\omega)$.

Пусть имеются составные предикаты 1-го, 2-го, ..., l -го уровней.

Выделим все часто встречающиеся (типичные) подформулы небольшой сложности формул $A_k^{l-1}(\omega)$ и обозначим их $P_i^l(x)$. Здесь x – список переменных, входящих в подформулу).

Обозначим предикаты, задаваемые этими подформулами посредством $p_i^l (i = 1, \dots, n_l)$ и будем называть их составными (сложными) предикатами l -го уровня.

Эти предикаты определяются соотношениями

$$p_i^l(x) \leftrightarrow P_i^l(x) \quad (7)$$

Формулы, полученные из $A_k^{l-1}(\omega)$ с помощью замены всех вхождений формул вида $P_i^l(x)$ на атомарные формулы $p_i^l(x)$, обозначим через $A_k^l(\omega)$. Такие формулы можно рассматривать как логические описания классов в терминах предикатов 1-го, 2-го, ..., l -го уровней.

Прекратить создание составных (сложных) предикатов очередного уровня можно в любой момент, но не позднее текущей ситуации, когда либо длины всех формул $A_k^l(\omega)$ меньше r , либо среди этих формул не найдется N подформул одинакового вида. В этом заключается правило остановки предлагаемого конечно-сходящегося алгоритма иерархического ОК.

В результате построения составных (сложных) предикатов и многоуровневого логического ОК исходная система ОК может быть записана с помощью равносильной ей многоуровневой логической системы ОК вида

$$\begin{array}{ccc}
 & A_k^L(\omega), & \\
 p_1^2(\omega) & \Leftrightarrow & P_1^2(\omega), \\
 & \vdots & \\
 p_{n_2}^2(\omega) & \Leftrightarrow & P_{n_2}^2(\omega), \\
 & \vdots & \\
 p_i^l(\omega) & \Leftrightarrow & P_i^l(\omega), \\
 & \vdots & \\
 p_{n_L}^L(\omega) & \Leftrightarrow & P_{n_L}^L(\omega).
 \end{array} \tag{8}$$

Можно показать, что при достаточно удачном выборе параметров r и N время работы алгоритмов, решающих различные задачи распознавания уменьшится [1].

3. Нейросетевое представление иерархического описания классов

При построении нейронных сетей для распараллеливания представлений логических ОК и распознавания сложных образов вычисление значений предикатов того или иного уровня может производиться соответствующим нейронным слоем или ядром много-ядерного компьютера. Так, например, при использовании только предикатов первого (исходного) и второго (синтезированного) уровней архитектура многослойной нейронной сети будет выглядеть так, как это представлено на рис. 1.

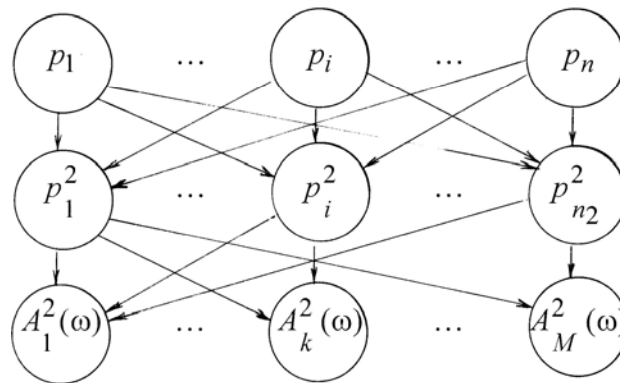


Рис.1. Архитектура многослойной нейронной сети

Таким образом, в первом слое (сенсорные нейроны) вычисляются все возможные значения исходных признаков распознаваемого объекта. Во втором слое вычисляются все возможные значения признаков второго уровня. В третьем слое проверяются формулы, задающие логические ОК.

В процессе распознавания образов сложный объект или его фрагмент будут отнесены к тому классу с номером k , для которого оказалась истинной формула $A_k^2(\omega)$.

Следует отметить, что процесс логического распознавания сложных образов на нейронной сети можно организовать таким образом, чтобы запоминать те фрагменты исходного объекта ω , для которых оказалась верна хотя бы одна из формул, определяющих принадлежность к классу. Тем самым может быть автоматически решена задача полного логического анализа сложного объекта.

Заключение

Таким образом, для сложных объектов (сложные изображения и сцены, смешанные сигналы и т.п.) предложены методы иерархического описания классов (образов) и оценки их эффективности в задачах логического анализа и распознавания образов. Описана информационная технология нейросетевого или много-ядерного представления иерархических описаний классов (образов), обеспечивающая массовый параллелизм при обработке данных и значительное ускорение процессов принятия решений в процессе распознавания образов.

Результаты получены при поддержке грантов РФФИ № 05-01-08044-офи-а и № 06-08-01612-а.

Список литературы

- [1]. Косовская Т.М. Оценки сложности решения задач распознавания при иерархическом описании классов – Тез. докл. на XI Междунар. конф. по проблемам теоретической кибернетики. Ульяновск, 1996.
- [2]. Косовская Т.М. Многоуровневые описания классов для принятия решений в задачах распознавания образов – Тр. III Междунар. конф. "Дискретные модели в теории управляющих систем". М., Диалог — МГУ, 1998.
- [3]. Тимофеев А.В. Роботы и искусственный интеллект. – М., "Наука", 1978. 191 с.
- [4]. Косовская Т.М., Тимофеев А.В. Об одном новом подходе к формированию логических решающих правил – Вестник ЛГУ, 1985, №8. С. 22–29. No 1, 160–176.
- [5]. Яблонский С.В. Тест – Математическая энциклопедия, т.5. М., 1985. С. 342–346. Publication. John Wiley & Sons, Inc. 2000. 491p.

- [6]. Тимофеев А.В., Каляев А.В. Методы обучения и минимизации сложности когнитивных нейромодулей супер-макро-нейрокомпьютеров с программируемой архитектурой. – Доклады Академии наук, 1994, т. 273, № 2, с. 180–183 (на русском и английском языках).
- [7]. Тимофеев А.В. Методы синтеза диофантовых нейросетей минимальной сложности. – Доклады Академии наук, 1995, т. 301, № 3, с. 1106–1109 (на русском и английском языках).
- [8]. Timofeev A.V., Semyonov A.V. Genetic Algorithms of Database Control and Knowledge Base Synthesis and Their Applications. – International Journal of Information Theories & Applications, Sofia, 1996, v.4, N1, pp. 17–22.
- [9]. Тимофеев А.В., Шеожев А. М., Шибзухов З. М. Мульти-агентные диофантовые нейронные сети в задачах распознавания и диагностики. – Нейрокомпьютеры: разработка и применение, 2005, № 10–11, с. 69–74.
-

Информация об авторах

Тимофеев Адиль Васильевич – докт. техн. наук, проф., Засл. деятель науки РФ, зав. лаб. информационных технологий в управлении и робототехнике СПИИРАН, зав. базовой кафедрой нейроинформатики и робототехники ГУАП; e-mail: tav@iias.spb.su

Косовская Татьяна Матвеевна – канд. физико-математических наук, доцент, докторант Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН); e-mail: kosov@NK1022.spb.edu