УДК 004.021
DOI: 10.24160/1993-6982-2019-5-129-134

О повышении эффективности вычислений при классификации изображений

А.И. Мамонтов

Рассмотрена автоматическая классификация изображений с помощью байесовского классификатора с целыми коэффициентами. Показано, как при сохранении качества классификации можно экономить память и уменьшать количество операций за счёт представления использующихся в классификаторе полиномов, отличающихся от общепринятых указанием только ненулевых коэффициентов.
Коэффициенты полиномов округляются и представляются с помощью целых чисел (чисел с фиксированной точкой). Проанализирован способ хранения полиномов, позволяющий хранить только ненулевые коэффициенты полинома (использующиеся при этом для классификации вектора и числа хранятся обычным способом).
Для экспериментов использована база данных MNIST. Для представления коэффициентов полиномов взято семь бит. При хранении коэффициентов без округлений нужно 11,77 Мбайт, при хранении округленных коэффициентов в обычном виде — 2,93 Мбайт, при хранении только ненулевых округленных коэффициентов — 1,3 Мбайт. Количество операций сложения и умножения, необходимых для принятия решения о классовой принадлежности изображения, значительно снизилось.
Дан экономный способ хранения матриц с коэффициентами полиномов на диске. В эксперименте с базой данных MNIST потребовалось лишь 0,35 Мбайт памяти для хранения полиномов в базе данных.
Результаты работы можно использовать при разработке программно-аппаратных средств для автоматической классификации изображений. Интересно также изучение аналогичных приёмов для повышения эффективности вычислений в других классификаторах.

Ключевые слова: классификация изображений, полиномы, целые числа, байесовский классификатор.

Для цитирования: Мамонтов А.И. О повышении эффективности вычислений при классификации изображений // Вестник МЭИ. 2019. № 5. С. 129—134. DOI: 10.24160/1993-6982-2019-5-129-134.

On Increasing the Computation Efficiency in Classifying Images

A.I. Mamontov

Automatic classification of images using the Bayes classifier with integer coefficients is considered. It is shown that by representing the polynomials used in the classifier that differ from the generally accepted ones through indicating only nonzero coefficients, it is possible to save memory and reduce the number of operations while keeping the same classification quality.
The polynomial coefficients are rounded and represented using integers (fixed-point numbers). A polynomial storing method that makes it possible to store only nonzero coefficients of the polynomial is analyzed. In using this method, the vectors and numbers used for carrying out classification are stored in the usual way.
The MNIST database was used for carrying out the experiments. Seven bits were taken to represent the polynomial coefficients. For storing coefficients without rounding, 11.77 MB of memory is needed; for storing the rounded coefficients in the usual form, 2.93 MB of memory is needed; and for storing only nonzero rounded coefficients, 1.3 MB is needed. The number of addition and multiplication operations required to make a decision about the image class has been reduced significantly.
An economical method for storing matrices with polynomial coefficients on a disk is presented. In the experiment with the MNIST database, only 0.35 MB of memory had to be used for storing the polynomials in the database.
The study results can be used in elaborating software and hardware for automatic classification of images. It is also of interest to study similar techniques to improve the efficiency of computations in other classifiers.

Key words: image classification, polynomials, integers, Bayesian classifier

For citation: Mamontov A.I. On Increasing the Computation Efficiency in Classifying Images. Bulletin of MPEI. 2019;5:129—134. (in Russian). DOI: 10.24160/1993-6982-2019-5-129-134.

Введение

Дана автоматическая классификация изображений с помощью байесовского классификатора с целыми коэффициентами [1, 2].

Методы организации вычислений в кольцах полиномов, а также методы представления знаний полиномами активно развиваются. В качестве примера можно использовать работы по схемам вычислений вещественных полиномов: Э.Г. Белага доказал невозможность построения схемы, требующей определенного числа операций [3], а В.Я. Пан построил схемы близкие к оптимальным [4]. Значительное количество публикаций посвящено определяющей скорость вычисления сложности полиномиальных представлений значных функций [5, 6].

При организации вычислений и представлении знаний часто возникают различные алгоритмы выделения общих подформул [7]. Представления полиномов с целыми коэффициентами описаны в [8 — 13].

Существуют работы о представлении с помощью линейных и нелинейных полиномов нейронных сетей — популярных конструкций, необходимых при классификации [14 — 16].

Перспективные модели распознавания с использованием полиномов с целыми коэффициентами рассматриваются в [2, 17 — 20].

В издании [2, с. 223] приведен график зависимости качества распознавания от размера используемых целых чисел. На нем проставлены параметры двойной точности, полученные в результате обучения байесовского классификатора и целочисленные параметры, полученные путем округления параметров двойной точности, а также с использованием разработанного в [2] алгоритма, основанного на методе ветвей и границ. Использован набор данных MNIST: 70 000 одинакового размера центрированных изображений рукописных цифр.

Рассмотрены задачи обоснования упомянутого выше способа представления используемых в байесовском классификаторе полиномов и экспериментального подтверждения эффективности подобных нестандартных представлений.

Дана классификация изображений с помощью байесовского классификатора, использующего полиномы с целыми коэффициентами. Приведены полиномы с целыми коэффициентами, позволяющими использовать меньше операций сложения и умножения и экономить память при сохранении той же функциональности, то есть повышать эффективность вычислений.

Доказано, что при использовании целочисленных моделей при распознавании [2, 17 — 20] можно изучать возможности для нестандартного представления полиномов для повышения эффективности вычислений.

О байесовском классификаторе для изображений

Задача классификации — формализованная задача, в которой имеется множество объектов, разделённых некоторым образом на классы. Задано конечное множество объектов (обучаюшая выборка), для которых известно, к каким классам они относятся. Принадлежность к классам остальных объектов неизвестна. Требуется на основе обучающей выборки построить алгоритм, способный классифицировать, то есть указать номер (или наименование) класса, к которому относится произвольный объект из исходного множества. Качество работы алгоритма можно оценивать с помощью различных мер. В настоящей работе качество работы алгоритма оценивается с помощью меры Accuracy — процента правильно классифицированных изображений из контрольной выборки.

Пусть обучающая выборка построена. Рассмотрим классификацию серых изображений размера $l \times l$ с помощью алгоритма, называемого байесовским классификатором. Определение номера (или наименования) класса, к которому относится изображение, сводится к нахождению номера функции, при котором максимально вычисляемое значение [1, с. 133—138]:

$$f_i(\mathbf{x}) = -\frac{n}{2}\ln 2\pi - \frac{1}{2}\ln\left|C_i\right| - \\ -\frac{1}{2}((\mathbf{x}-\mathbf{m}_i)C_i^{-1}\left(\mathbf{x}-\mathbf{m}_i\right)^T) + \ln P_i,\ i = 1,\ldots,k,$$

где $\mathbf{x}$ — вектор параметров контрольного изображения размера $n = l \times l$ — яркости всех пикселей серого изображения; k — количество классов, к которым может принадлежать изображение; $\mathbf{m}_i$, C_i — математическое ожидание изображения и симметричная ковариационная матрица для изображений i-го класса обучающей выборки; P_i — вероятность того, что случайное изображение обучающей выборки принадлежит i-му классу.

Заметим, что C_i — матрицы размера n на n; $\mathbf{m}_i$ — вектора размера n, а P_i — числа.

Обозначим

$$A^{(i)} = -\frac{1}{2}C_i^{-1};\ \ B^{(i)} = -\frac{n}{2}\ln 2\pi - \frac{1}{2}\ln\left|C_i\right| + \ln P_i,\ \ i = 1,\ldots,k,$$

тогда

$$f_i(\mathbf{x}) = \left(\mathbf{x}-\mathbf{m}_i\right)A^{(i)}\left(\mathbf{x}-\mathbf{m}_i\right)^T + B^{(i)},\ \ i = 1,\ldots,k.$$

Матрицы $A^{(i)}$ симметричны, так как обратны к симметричным матрицам.

Рассмотрим

$$A^{(i)} = (a_{\alpha\beta}^{(i)})_{\alpha=1,\beta=1}^{n,n};$$
$$\mathbf{t}_i = \mathbf{x} - \mathbf{m}_i = \left(t_{i,1},\ \ t_{i,2},\ ...,\ t_{i,n}\right)$$

и преобразуем $f_i(\mathbf{x})$:

$$f_i(\mathbf{x}) = \mathbf{t}_i A^{(i)} \mathbf{t}_i^T + B^{(i)} = \\ = \left(t_{i,1} \quad t_{i,2} \quad \ldots \quad t_{i,n}\right)\times$$

$$\times\begin{pmatrix} a_{11}^{(i)} & a_{12}^{(i)} & \cdots & a_{1n}^{(i)} \\ a_{21}^{(i)} & a_{22}^{(i)} & \cdots & a_{2n}^{(i)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1}^{(i)} & a_{n2}^{(i)} & \cdots & a_{nn}^{(i)} \end{pmatrix}\begin{pmatrix} t_{i,1} \\ t_{i,2} \\ \cdots \\ t_{i,n} \end{pmatrix} + B^{(i)} =$$

$$= a_{11}^{(i)}t_{i,1}t_{i,1} + a_{12}^{(i)}t_{i,1}t_{i,2} + \cdots + a_{1n}^{(i)}t_{i,1}t_{i,n} +$$
$$+ a_{21}^{(i)}t_{i,2}t_{i,1} + a_{22}^{(i)}t_{i,2}t_{i,2} + \cdots + a_{2n}^{(i)}t_{i,2}t_{i,n} +$$
$$+\ldots+$$
$$+ a_{n1}^{(i)}t_{i,n}t_{i,1} + a_{n2}^{(i)}t_{i,n}t_{i,2} + \cdots + a_{nn}^{(i)}t_{i,n}t_{i,n} + B^{(i)}.$$

Коэффициенты $a_{\alpha\beta}^{(i)}$ округлим и представим целыми числами (числами с фиксированной точкой). Подобный приём применяется также в [2, 17 — 20]. Чтобы не вводить новые обозначения, округленные матрицы $A^{(i)}$ обозначим как $A^{(i)} = (a_{\alpha\beta}^{(i)})_{\alpha=1,\beta=1}^{n,n}$. Далее используем только матрицы $A^{(i)}$ с округленными коэффициентами.

Многие коэффициенты матриц $A^{(i)}$ являются малыми по абсолютной величине числами. Это происходит в том случае, когда часть изображения составляет белый фон, возникающий после удаления основного фона. Рассмотрим способ представления матриц в данной задаче.

Способ представления и хранения матриц при байесовской классификации

Создадим списки списков ненулевых коэффициентов матриц $A^{(i)}$. Для этого при $i = 1,\ldots, k$ построим списки:

— $A^{(1, i)}$ — списков элементов β таких, что $a_{\alpha\beta}^{(i)} = 1$, $\alpha \le \beta$, $\alpha = 1, \ldots, n$;

— $A^{(-1, i)}$ — списков элементов β таких, что $a_{\alpha\beta}^{(i)} = -1$, $\alpha \le \beta$, $\alpha = 1, \ldots, n$;

— $A^{(2, i)}$ — списков элементов β таких, что $a_{\alpha\beta}^{(i)} = 2$, $\alpha \le \beta$, $\alpha = 1, \ldots, n$;

— $A^{(-2, i)}$ — списков элементов β таких, что $a_{\alpha\beta}^{(i)} = -2$, $\alpha \le \beta$, $\alpha = 1, \ldots, n$;

— $A^{(*, i)}$ — списков элементов $<\beta, a_{\alpha\beta}^{(i)}>$ таких, что $|a_{\alpha\beta}^{(i)}| > 2$, $\alpha \le \beta$, $\alpha = 1, \ldots, n$.

Пример 1.

Для матрицы

$$A^{(i)} = \begin{pmatrix} 1 & 0 & -2 & 1 & 0 & 0 \\ 0 & 2 & 6 & -1 & -2 & 0 \\ -2 & 6 & 1 & -1 & 0 & -1 \\ 1 & -1 & -1 & 2 & -1 & 1 \\ 0 & -2 & 0 & -1 & 7 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \end{pmatrix}, n = 6$$

построим $A^{(1, i)}$, учтем $a_{\alpha\beta}^{(i)} = 1$, $\alpha \le \beta$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

и получим $A^{(1, i)} = ((1,4), (), (3), (6), (), (6))$.

Действуя аналогично, получим:

$$A^{(-1, i)} = ((), (4), (4,6), (5), (6) ,());$$
$$A^{(2, i)}) = ((), (2), (), (4), (), ());$$
$$A^{(-2, i)} = ((3), (5), (), (), (), ());$$
$$A^{(*, i)} = ((), (<3,6>), (), (), (<5,7>), ()).$$

При таком представлении не нужно хранить нулевые коэффициенты. Векторы $\mathbf{m}_i$ и свободные члены $B^{(i)}$ хранятся обычным способом.

При рассматриваемом варианте представления данных при вычислении $f_i(\mathbf{x})$ можно уменьшать количество сложений и умножений: не складывать в общей формуле $a_{\alpha\beta}^{(i)} t_{i,\alpha} t_{i,\beta}$, где коэффициент $a_{\alpha\beta}^{(i)} = 0$, и не умножать многократно на 1, –1, 2, –2. При этом количество операций сложений и умножений и требуемой памяти составляет $O(r)$, где r — количество всех нулевых коэффициентов во всех матрицах $A^{(i)}$, а не $O(n^2)$, как при представлении полиномов с помощью квадратных матриц.

Для экспериментов использована база данных MNIST. Для обучения взяты 5000 случайных изображений из этой коллекции, а для распознавания — другие 5000 случайных изображений. Для представления коэффициентов $a_{\alpha\beta}^{(i)}$ применены семибитовые числа. При хранении коэффициентов без округлений нужно 11,77 Мб, округленных коэффициентов в виде обычной матрицы и вектора — 2,93 Мб, только ненулевых округленных коэффициентов — 1,3 Мб. Скорость вычисления $f_i(\mathbf{x})$ увеличилась в 6 раз.

Способ хранения на диске совокупности матриц

Совокупность матриц $A^{(i)}$ можно экономно хранить на диске в виде битовых списков (нулей и единиц) и списков целых чисел. Для описания алгоритма их построения рассмотрим процедуру, работающую с целыми числами и битовым списком.

Процедура БИТСП(a, b, d, l_d) получает на входе целые числа a, b, список нулей и единиц d и длину этого списка l_d. Если $a = b$, то в результате работы процедуры БИТСП к битовому списку d добавляется 0, а если $a \ne b$, то 1, длина списка d увеличивается на 1.

Рассмотрим алгоритм построения битовых списков ($\tilde{d}$, $d^{(0, i)}$, $d^{(1, i)}$, $d^{(-1, i)}$, $d^{(2, i)}$, $d^{(-2, i)}$) длиной $\tilde{l}$, l_{0i}, l_{1i}, l_{-1i}, l_{2i}, l_{-2i} и списков целых чисел $d^{(*, i)}$ длиной l_{*i}, $i = 1,\ldots, k$:

1. Создать пустые списки $\tilde{d}$, $d^{(0,\,i)}$, $d^{(1,\,i)}$, $d^{(-1,\,i)}$, $d^{(2,\,i)}$, $d^{(-2,\,i)}$), $d^{(*,\,i)}$ при $i = 1,\ldots, k$.
2. Положить $\tilde{l} = 0$; $l_{0i} = 0$; $l_{1i} = 0$; $l_{-1i} = 0$; $l_{2i} = 0$; $l_{-2i} = 0$; $l_{*i} = 0$ при $i = 1,\ldots, k$.
3. Для $\alpha = 1,\ldots, n$ выполнить действия:
 3.1 Для $\beta = \alpha,\ldots, n$ выполнить действия:
  Присвоить $\tilde{l} = \tilde{l} + 1$
  Если $a^{(i)}_{\alpha\beta} = 0$ при всех $i = 1, \ldots, k$, то
   Положить $\tilde{d}_{\tilde{l}} = 0$
  Иначе
   Положить $\tilde{d}_{\tilde{l}} = 1$
   Для $i = 1,\ldots, k$ выполнить:
    БИТСП($a^{(i)}_{\alpha\beta}$, 0, $d^{(0,\,i)}$, l_{0i})
    Если $a^{(i)}_{\alpha\beta} \neq 0$, то выполнить:
     БИТСП($a^{(i)}_{\alpha\beta}$, 1, $d^{(1,\,i)}$, l_{1i})
     Если $a^{(i)}_{\alpha\beta} \neq 1$, то выполнить:
      БИТСП($a^{(i)}_{\alpha\beta}$, –1, $d^{(-1,\,i)}$, l_{-1i})
      Если $a^{(i)}_{\alpha\beta} \neq -1$, то выполнить:
       БИТСП($a^{(i)}_{\alpha\beta}$, 2, $d^{(2,\,i)}$, l_{2i})
       Если $a^{(i)}_{\alpha\beta} \neq -2$, то выполнить:
        БИТСП($a^{(i)}_{\alpha\beta}$, –2, $d^{(-2,\,i)}$, l_{-2i})
        Если $a^{(i)}_{\alpha\beta} \neq 2$, то выполнить:
        Присвоить $l_{*i} = l_{*i} + 1$
        Положить $d^{(*,i)}_{l*_i} = a^{(i)}_{\alpha\beta}$

Пример 2. Пусть $k = 2$.

$$A^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{pmatrix};\; A^{(2)} = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

тогда верхние треугольники этих матриц составят:

$$\begin{pmatrix} 0 & 1 & 0 \\ * & 1 & 2 \\ * & * & 0 \end{pmatrix};\; \begin{pmatrix} 0 & 1 & 4 \\ * & 1 & 0 \\ * & * & 0 \end{pmatrix},$$

и $\tilde{d} = (0, 1, 1, 1, 1, 0)$; $d^{(0,\,1)} = (1, 0, 1, 1)$; $d^{(1,\,1)} = (0, 0, 1)$; $d^{(-1,\,1)} = (1)$; $d^{(2,\,1)} = (0)$; $d^{(-2,\,1)} = ()$; $d^{(*,\,1)} = ()$; $d^{(0,\,2)} = (1, 1, 1, 0)$; $d^{(1,\,2)} = (0, 1, 0)$; $d^{(-1,\,2)} = (0)$; $d^{(2,\,2)} = (0)$; $d^{(-2,\,2)} = (0)$; $d^{(*,\,2)} = (4)$.

Для хранения совокупности матриц на диск достаточно последовательно записать битовый список $\tilde{d}$, для каждого $i = 1, \ldots, k$ следующие друг за другом битовые списки $d^{(0,\,i)}$, $d^{(1,\,i)}$, $d^{(-1,\,i)}$, $d^{(2,\,i)}$, $d^{(-2,\,i)}$ и список целых чисел $d^{(*,\,i)}$.

Алгоритм применяют как к совокупности матриц $A^{(i)}$, так и к совокупности списков $A^{(1,\,i)}$, $A^{(-1,\,i)}$, $A^{(2,\,i)}$, $A^{(-2,\,i)}$, $A^{(*,\,i)}$.

Для загрузки совокупности списков $A^{(1,\,i)}$, $A^{(-1,\,i)}$, $A^{(2,\,i)}$, $A^{(-2,\,i)}$, $A^{(*,\,i)}$ с диска можно применить алгоритм, аналогичный изложенному выше.

Обозначим верхнюю треугольную часть симметричных матриц $A^{(i)}$ с помощью $U^{\,i)}$:

R_0 — количество всех коэффициентов, равных нулю, в каждой из матриц $U^{(i)}$;

r_0 — количество всех нулевых коэффициентов во всех матрицах $U^{(i)}$;

r_1 — количество всех единичных коэффициентов во всех матрицах $U^{(i)}$;

r_{-1} — количество всех коэффициентов, равных –1, во всех матрицах $U^{(i)}$;

r_2 — количество всех коэффициентов, равных 2, во всех матрицах $U^{(i)}$;

r_{-2} — количество всех коэффициентов, равных –2, во всех матрицах $U^{(i)}$;

r_3 — количество всех коэффициентов, по модулю больших 2, во всех матрицах $U^{(i)}$.

Памяти требуется:

$$\begin{aligned} & l^4 + (kl^4 - kR_0) + \left(kl^4 - r_0\right) + \\ & + \left(kl^4 - r_0 - r_1\right) + (kl^4 - r_0 - r_1 - \\ & - r_{-1}) + (kl^4 - r_0 - r_1 - r_{-1} - r_2) + 7r_3 \text{ бит.} \end{aligned}$$

В эксперименте с базой данных MNIST потребовалось 0,35 Мб памяти для хранения на диске данных всех возникающих после обучения: коэффициентов матриц $A^{(i)}$, векторов $\mathbf{m}_i$ и свободных членов $B^{(i)}$. Напомним, что при обычном способе хранения коэффициентов нужно 11,77 Мб.

Заключение

Показано, что при сохранении функциональности можно использовать меньше операций сложения и умножения и экономить память при классификации изображений с помощью байесовского классификатора с целыми коэффициентами, то есть повысить эффективность вычислений. Результаты работы полезны при разработке программно-аппаратных средств для автоматической классификации изображений. Также может быть интересно изучение аналогичных приёмов для повышения эффективности вычислений в других классификаторах.

Литература

1. **Ту Дж. Р., Гонсалес Р.** Принципы распознавания образов. М.: Мир, 1978.
2. **Tschiatschek S., Paul K., Pernkopf F.** Integer Bayesian Network Classifiers // Machine Learning and Knowledge Discovery in Databases. Berlin, Heidelberg: Springer, 2014. Pp. 209—224.

References

1. **Tu Dzh. R., Gonsales R.** Printsipy raspoznavaniya obrazov. M.: Mir, 1978. (in Russian).
2. **Tschiatschek S., Paul K., Pernkopf F.** Integer Bayesian Network Classifiers. Machine Learning and Knowledge Discovery in Databases. Berlin, Heidelberg: Springer, 2014:209—224.

3. **Белага Э.Г.** О вычислении значений многочлена от одного переменного с предварительной обработкой коэффициентов // Проблемы кибернетики. М.: Физматгиз, 1961. Вып. 5. С. 7—15.

4. **Пан В.Я.** Некоторые схемы для вычисления значений полиномов с вещественными коэффициентами // Там же. С. 17—29.

5. **Селезнева С.Н.** Сложность систем функций алгебры логики и систем функций трехзначной логики в классах поляризованных полиномиальных форм // Дискретная математика. 2015. Т. 27. Вып. 1. С. 111—122.

6. **Маркелов Н.К.** Нижняя оценка сложности функций трехзначной логики в классе поляризованных полиномов // Вестник Московского университета. Серия 15. «Вычислительная математика и кибернетика». 2012. Вып. 3. С. 40—45.

7. **Косовская Т.М.** Самообучающаяся сеть с ячейками, реализующими предикативные формулы // Труды СПИИРАН. 2015. №. 6. С. 94—113.

8. **Алексиадис Н.Ф.** Алгоритмическая неразрешимость проблемы полноты для полиномов с целыми коэффициентами // Вестник МЭИ. 2015. № 3. С. 110—117.

9. **Алексиадис Н.Ф.** Алгоритмическая неразрешимость задачи о нахождении базиса конечной полной системы полиномов с целыми коэффициентами // Интеллектуальные системы. Теория и приложения. 2016. Т. 20. Вып. 3. С. 19—23.

10. **Мамонтов А.И., Мещанинов Д.Г.** Проблема полноты в функциональной системе линейных полиномов с целыми коэффициентами // Дискретная математика. 2010. Т. 22. № 4. С. 64—82.

11. **Мамонтов А.И., Мещанинов Д.Г.** Алгоритм распознавания полноты в функциональной системе $L(Z)$ // Дискретная математика. 2014. Т. 26. № 1. С. 85—95.

12. **Мамонтов А.И.** Об использующем суперпозиции способе эффективного вычисления систем линейных полиномов с целыми коэффициентами // Интеллектуальные системы. Теория и приложения. 2016. Т. 20. Вып. 3. С. 58—63.

13. **Мамонтов А.И., Рябинов С.М.** Об одном методе экономии памяти при классификации текстов // Программные системы: теория и приложения. 2017. Т. 8. № 4 (35). С. 133—147.

14. **Gorban A.N., Wunsch D.C.** The General Approximation Theorem // Proc. IEEE IJCNN'98. 1998. Pp. 1271—1274.

15. **Горбань А.Н.** Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики. 1998. Т. 1. № 1. С. 11—24.

16. **Половников В.С.** О нелинейных характеристиках нейронных схем в произвольных базисах // Интеллектуальные системы. 2013. Т. 17. № 1—4. С. 87—90.

17. **Tschiatschek S., Pernkopf F.** On Bayesian Network Classifiers with Reduced Precision Parameters //

3. **Belaga E.G.** O Vychislenii Znacheniy Mnogochlena ot Odnogo Peremennogo s Predvaritel'noy Obrabotkoy Koeffitsientov. Problemy Kibernetiki. M.: Fizmatgiz, 1961;5:7—15. (in Russian).

4. **Pan V.Ya.** Nekotorye Skhemy dlya Vychisleniya Znacheniy Polinomov s Veshchestvennymi Koeffitsientami. Tam zhe:17—29. (in Russian).

5. **Selezneva S.N.** Slozhnost' Sistem Funktsiy Algebry Logiki i Sistem Funktsiy Trekhznachnoy Logiki v Klassah Polyarizovannyh Polinomial'nyh Form. Diskretnaya Matematika. 2015;27;1:111—122. (in Russian).

6. **Markelov N.K.** Nizhnyaya Otsenka Slozhnosti Funktsiy Trekhznachnoy Logiki v Klasse Polyarizovannyh Polinomov. Vestnik Moskovskogo Universiteta. Seriya 15. «Vychislitel'naya Matematika I Kibernetika». 2012;3: 40—45. (in Russian).

7. **Kosovskaya T.M.** Samoobuchayushchayasya Set' s Yacheykami, Realizuyushchimi Predikativnye Formuly. Trudy SPIIRAN. 2015;6:94—113. (in Russian).

8. **Aleksiadis N.F.** Algoritmicheskaya Nerazreshimost' Problemy Polnoty dlya Polinomov s Tselymi Koeffitsientami. Vestnik MEI. 2015;3:110—117. (in Russian).

9. **Aleksiadis N.F.** Algoritmicheskaya Nerazreshimost' Zadachi o Nahozhdenii Bazisa Konechnoy Polnoy Sistemy Polinomov s Tselymi Koeffitsientami. Intellektual'nye Sistemy. Teoriya I Prilozheniya. 2016;20;3:19—23. (in Russian).

10. **Mamontov A.I., Meshchaninov D.G.** Problema Polnoty v Funktsional'noy Sisteme Lineynyh Polinomov s Tselymi Koeffitsientami. Diskretnaya Matematika. 2010; 22;4:64—82. (in Russian).

11. **Mamontov A.I., Meshchaninov D.G.** Algoritm Raspoznavaniya Polnoty v Funktsional'noy Sisteme $L(Z)$. Diskretnaya Matematika. 2014;26;1:85—95. (in Russian).

12. **Mamontov A.I.** Ob Ispol'zuyushchem Superpozitsii Sposobe Effektivnogo Vychisleniya Sistem Lineynyh Polinomov s Tselymi Koeffitsientami. Intellektual'nye Sistemy. Teoriya i Prilozheniya. 2016;20; 3:58—63. (in Russian).

13. **Mamontov A.I., Ryabinov S.M.** Ob Odnom Metode Ekonomii Pamyati pri Klassifikatsii Tekstov. Programmnye Sistemy: Teoriya I Prilozheniya. 2017;8;4 (35): 133—147. (in Russian).

14. **Gorban A.N., Wunsch D.C.** The General Approximation Theorem. Proc. IEEE IJCNN'98. 1998:1271—1274.

15. **Gorban' A.N.** Obobshchennaya approksimatsionnaya Teorema i Vychislitel'nye Vozmozhnosti Neyronnyh Setey. Sibirskiy Zhurnal vychislitel'noy matematiki. 1998;1;1:11—24. (in Russian).

16. **Polovnikov V.S.** O Nelineynyh Kharakteristikah Neyronnyh Skhem v Proizvol'nyh Bazisah. Intellektual'nye Sistemy. 2013;17;1—4:87—90. (in Russian).

17. **Tschiatschek S., Pernkopf F.** On Bayesian Network Classifiers with Reduced Precision Parameters. IEEE

IEEE Trans. Pattern Analysis and Machine Intelligence. 2015. V. 37 (4). Pp. 774—785.

18. **Yi Y., Hangping Z., Bin Z.** A New Learning Algorithm for Neural Networks with Integer Weights and Quantized Non-linear Activation Functions // Artificial Intelligence in Theory and Practice II. Boston: Springer, 2008. V. 276. Pp. 427—431.

19. **Anguita D., Ghio A., Pischiutta S., Ridella S.** A Support Vector Machine with Integer Parameters // Neurocomputing. 2008. V. 72. No. 1—3. Pp. 480—489.

20. **Shuang Wu, Guoqi Li, Feng Chen, Luping Shi.** Training and Inference with Integers in Deep Neural Networks // Proc. ICLR. 2018. Pp. 1—14.

Trans. Pattern Analysis and Machine Intelligence. 2015; 37 (4):774—785.

18. **Yi Y., Hangping Z., Bin Z.** A New Learning Algorithm for Neural Networks with Integer Weights and Quantized Non-linear Activation Functions. Artificial Intelligence in Theory and Practice II. Boston: Springer, 2008;276:427—431.

19. **Anguita D., Ghio A., Pischiutta S., Ridella S.** A Support Vector Machine with Integer Parameters. Neurocomputing. 2008;72;1—3:480—489.

20. **Shuang Wu, Guoqi Li, Feng Chen, Luping Shi.** Training and Inference with Integers in Deep Neural Networks. Proc. ICLR. 2018:1—14.

Сведения об авторе:

Мамонтов Андрей Игоревич — кандидат технических наук, доцент кафедры математического моделирования НИУ «МЭИ», e-mail: MamontovAI@yandex.ru

Information about author:

Mamontov Andrey I. — Ph.D. (Techn.), Assistant Professor of Mathematical Modeling Dept., NRU MPEI, e-mail: MamontovAI@yandex.ru

Работа выполнена при поддержке: РФФИ (проект № 17-01-00485а)
The work is executed at support: RFBR (project No. 17-01-00485a)

Статья поступила в редакцию: *08.11.2018*
The article received to the editor: *08.11.2018*