УДК 004.021
DOI: 10.24160/1993-6982-2021-3-103-109

О способах экономии компьютерной памяти при классификации данных с помощью полносвязных нейронных сетей

А.И. Мамонтов

При решении задачи классификации в качестве математической модели взята полносвязная обучаемая (настраиваются параметры, представляемые вещественными числами двойной точности) нейронная сеть. После того как обучение проведено, ее параметры округляются и представляются числами с фиксированной точкой (целыми числами). Цель исследования — сокращение требуемого объёма памяти вычислительной системы для хранения полученных целочисленных параметров.
Для минимизации объёма памяти разрабатываются следующие способы хранения целочисленных параметров, основанные на представлении входящих в полносвязную нейронную сеть линейных полиномов с помощью композиций более простых функций:
— способ, базирующийся на представлении полинома в виде суммы более простых полиномов;
— способ, опирающийся на отдельное хранение информации о сложениях и умножениях.
В эксперименте с набором данных MNIST для хранения вещественных параметров полносвязной нейронной сети потребовалось 1,41 Мб, для хранения целочисленных параметров без применения предлагаемых способов — 0,7 Мб, с применением первого способа — 0,47 Мб в ОЗУ и 0,3 Мб в сжатом виде на диске, с применением второго способа — 0,25 Мб на диске.
В эксперименте с набором данных USPS для хранения вещественных параметров полносвязной нейронной сети потребовалось 0,25 Мб, для хранения целочисленных параметров без применения разрабатываемых способов — 0,1 Мб, с применением первого способа — 0,05 Мб в ОЗУ и примерно столько же в сжатом виде на диске, с применением второго способа — 0,03 Мб на диске.
Результаты работы могут быть применены при использовании полносвязных нейронных сетей для решения различных задач распознавания в условиях аппаратных ограничений.

Ключевые слова: линейные полиномы, целые числа, полносвязная нейронная сеть.

Для цитирования: Мамонтов А.И. О способах экономии компьютерной памяти при классификации данных с помощью полносвязных нейронных сетей // Вестник МЭИ. 2021. № 3. С. 103—109. DOI: 10.24160/1993-6982-2021-3-103-109.

On Computer Memory Saving Methods in Performing Data Classification Using Fully Connected Neural Networks

A.I. Mamontov

In solving the classification problem, a fully connected trainable neural network (with adjusting the parameters represented by double-precision real numbers) is used as a mathematical model. After the training is completed, the neural network parameters are rounded and represented as fixed-point numbers (integers). The aim of the study is to reduce the required amount of the computing system memory for storing the obtained integer parameters.
To reduce the amount of memory, the following methods for storing integer parameters are developed, which are based on representing the linear polynomials included in a fully connected neural network using compositions of simpler functions:
— a method based on representing the considered polynomial as a sum of simpler polynomials;
— a method based on separately storing the information about additions and multiplications.
In the experiment with the MNIST data set, it took 1.41 MB to store real parameters of a fully connected neural network, 0.7 MB to store integer parameters without using the proposed methods, 0.47 MB in the RAM and 0.3 MB in compressed form on the disk when using the first method, and 0.25 MB on the disk when using the second method.
In the experiment with the USPS data set, it took 0.25 MB to store real parameters of a fully connected neural network, 0.1 MB to store integer parameters without using the proposed methods, 0.05 MB in the RAM and approximately the same amount in compressed form on the disk when using the first method, and 0.03 MB on the disk when using the second method. The study results can be applied in using fully connected neural networks to solve various recognition problems under the conditions of limited hardware capacities.
Key words: linear polynomials, integers, fully connected neural network.

For citation: Mamontov A.I. On Computer Memory Saving Methods in Performing Data Classification Using Fully Connected Neural Networks. Bulletin of MPEI. 2021;3:103—109. (in Russian). DOI: 10.24160/1993-6982-2021-3-103-109.

Введение

Задачи автоматической классификации могут решаться в условиях аппаратных ограничений, например, на встроенных или портативных вычислительных платформах, которые становятся все более мощными. В публикациях [1, 2] упоминаются примеры подобных задач: классификация слуховых сцен в слуховых аппаратах, вычисления в спутниках и сенсорных сетях. Для их решения авторы [1, 2] предлагают использовать числа с фиксированной точкой (представляемые с помощью целых чисел). В наших исследованиях мы также используем вычисления с целыми числами и рассмотрим задачу экономии памяти вычислительной системы, возникающую в условиях аппаратных ограничений.

Вопросы экономии памяти рассматриваются при решении задач распознавания. Применяются коды Хафмана [3] и разреженные матрицы [4]. В настоящей работе проанализированы способы экономии памяти, основанные на том, что функции, входящие в используемую для решения задачи классификации математическую модель, представляются с помощью композиций (суперпозиций) более простых функций.

Для решения задачи классификации использована полносвязная нейронная сеть. Нейронные сети — современные и достаточно распространённые модели, применяемые при решении различных задач из области распознавания. Работы по нейронным сетям достаточно распространены в научной среде. Научный интерес представляет [5] о портировании в микросхему использующей вычисления с фиксированной точкой нейронной сети для классификации цифр.

Полносвязная нейронная сеть интерпретируется совокупностью формул, среди которых выделим линейные полиномы, коэффициенты которых округлим и представим целыми числами. Получающиеся целочисленные линейные полиномы выразим с помощью композиций (суперпозиций) более простых линейных полиномов. Представление полиномов с целыми коэффициентами с помощью композиций других полиномов изложено в [6 — 10]. Авторы [11 — 13] изучали представление с помощью линейных и нелинейных полиномов искусственных нейронных сетей.

Отметим также работы, в которых описаны перспективные целочисленные модели распознавания. Так, в [1, 14 — 17] рассмотрен метод опорных векторов с целочисленными параметрами, в [2, 18] исследован байесовский классификатор с целочисленными параметрами, а в [19, 20] — искусственные нейронные сети с целочисленными параметрами.

Постановка задачи

Задача классификации — формализованная задача, в которой имеется множество объектов, разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Множество разделено на два подмножества: обучающую и контрольную выборки. Принадлежность к классам остальных объектов неизвестна. Требуется на основе обучающей выборки построить алгоритм, способный классифицировать, т. е. указать номер класса, к которому относится произвольный объект из исходного множества. Качество работы алгоритма оценим с помощью метрики *Accuracy* — процента правильно классифицированных объектов из контрольной выборки.

Возьмем уже построенный на основе обучающей выборки алгоритм классификации данных и разработаем с его помощью способы экономии памяти вычислительной системы при классификации данных.

При классификации данных с помощью полносвязных нейронных сетей номер класса, к которому принадлежит объект

$$\mathbf{x}_1 = \begin{pmatrix} x_{1,1} \\ \dots \\ x_{1,n1} \end{pmatrix} \in \mathbf{R}^{n1},$$

указывается с помощью построенного на основе обучающей выборки алгоритма.

Для $i = 1, \dots, s$ выполним вычисления по формулам:

$$\begin{aligned} y_{i,j} &= a_{i,j,0} + \sum_{k=1}^{n_i} a_{i,j,k} x_{i,k}, \quad j = 1, 2, \dots, m_i; \\ x_{i+1,t} &= f_{i,t}\left(y_{i,1}, y_{i,2}, \dots, y_{i,mi}\right), \quad t = 1, 2, \dots, n_{i+1}, \end{aligned} \qquad (1)$$

где s, n_i, m_i — натуральные числа, причём $n_{s+1} = 1$; $f_{i,\,t}$ — нелинейные вещественные функции; $a_{i,\,j,k}$ — вещественные числа.

Следовательно, при каждом i (в каждом слое сети), вычисляются m_i значений линейных полиномов $y_{i,\,j}$, затем значения $y_{i,\,j}$ становятся аргументами нелинейных вещественных функций (функций активации), которые возвращают вещественные значения $x_{i+1,t}$ — исходные данные для следующей ($i + 1$)-й итерации (следующего слоя). Значение $x_{s+1,t}$ — единственное, это номер класса, указанный алгоритмом.

Коэффициенты $a_{i,j,k}$ округлим до определенного знака и представим целыми числами (числами с фиксированной точкой). Чтобы не вводить новые обозначения, округлённые коэффициенты $a_{i,\,j,k}$ и сами формулы (1), в которых эти коэффициенты применяются, продолжим обозначать $a_{i,\,j,k}$ и (1). Используем только округленные коэффициенты $a_{i,\,j,k}$.

Экспериментально влияние округления на качество классификации оценивается в ряде работ [1, 2, 5, 14, 17, 20], влияние округления на качество классификации аналитически исследовано в [15, 16]. В настоящей работе влияние округления на качество классификации оценим экспериментально, но напомним некоторые утверждения о вычислительных процессах, возникающих при суммировании чисел. Они, на наш взгляд, полезны для понимания, почему ошибка округления может не очень сильно влиять на качество классификации.

Утверждение 1.

Пусть $\alpha_1, \ldots, \alpha_n$ *— вещественные числа известные точно, а* $\xi_1, \ldots, \xi_n$ *— приближённые вещественные числа. Тогда для границы абсолютной погрешности значения* $S^* = \sum_{i=1}^{n} \alpha_i \xi_i$ *справедлива следующая оценка*

$$\overline{\Delta}\left(S^*\right) \approx \sum_{i=1}^{n} \left|\alpha_i\right| \overline{\Delta}\left(\xi_i^*\right),$$

где $\overline{\Delta}\left(\xi_i^*\right)$ — границы абсолютных погрешностей переменных, $i = 1, \ldots, n$.

Утверждение 2 (условие Линдеберга).

Пусть $\alpha_1, \ldots, \alpha_n, \ldots$ *— вещественные числа известные точно,* $\xi_1, \ldots, \xi_n, \ldots$ *— независимые вещественные случайные величины, имеющие конечные математические ожидания и дисперсии. Рассмотрим линейную комбинацию*

$$\sum_{i=1}^{n} \alpha_i \xi_i .$$

Определим новое множество переменных $\zeta_i = \alpha_i \xi_i$ *со следующими математическии ожиданиями и дисперсиями:* $\mu_{\zeta i} = \alpha_i \mu_{\xi i}$ и $\sigma_{\zeta i}^2 = \alpha_i^2 \sigma_{\xi i}^2$. *Тогда, если ни одно из* $\sigma_{\zeta i}^2$ *не доминирует над суммой дисперсий* $s_n^2 = \sum_{i=1}^{n} \sigma_{\zeta i}^2$, *т. е.*

$$\forall \varepsilon > 0 \lim_{n \to \infty} \sum_{i=1}^{n} M \left(\frac{\left(\zeta_i - \mu_{\zeta i}\right)^2}{s_n^2} E\left(\left|\zeta_i - \mu_{\zeta i}\right| > \varepsilon s_n\right) \right) = 0,$$

где $E\left(\left|\zeta_i - \mu_{\zeta i}\right| > \varepsilon s_n\right) = \begin{cases} 1, \left|\zeta_i - \mu_{\zeta i}\right| > \varepsilon s_n; \\ 0, \text{иначе} \end{cases}$; *M — математическое ожидание, то переменная*

$$S_n = \frac{\sum_{i=1}^{n} \left(\zeta_i - \mu_{\zeta i}\right)}{s_n}$$

стремится к стандартному нормальному распределению.

Таким образом, абсолютную погрешность $\sum_{i=1}^{n} \alpha_i \xi_i$ при соблюдении определённых условию можно оценивать с помощью центральных предельных теорем.

Пример 1. Требуется сложить 10000 чисел, каждое из которых округлено с точностью до 10^{-5}. Предположим, что ошибки, возникающие при округлении чисел, независимы и равномерно распределены в интервале $(-0{,}5 \cdot 10^{-5}; 0{,}5 \cdot 10^{-5})$. Найдем границы, в которых с вероятностью 0,99 будет лежать суммарная ошибка. *Ответ*: суммарная ошибка округления десяти тысяч чисел с вероятностью не менее 0,99 лежит в границах $(-74 \cdot 10^{-5}; 74 \cdot 10^{-5})$.

Вполне может получиться, что ошибка округления не сильно повлияет на качество классификации, как в случае малых (утверждение 1), так и в случае больших *n*.

Задача

Рассмотрим нейронные сети (1), в которых велико количество нулевых коэффициентов $a_{i,\,j,k}$, и создадим способы их хранения, позволяющие экономить память вычислительной системы.

Метод решения

Способы хранения коэффициентов $a_{i,j,k}$ будем строить, основываясь на представлении линейных полиномов системы (1) с помощью композиций более простых функций.

Найдем некоторые простые части в линейных полиномах системы (1) и представим их в виде композиций простых частей.

Следует отметить, что подобные представления сохраняют возможность вычисления линейных полиномов.

Используем следующие варианты.

Систему линейных полиномов выразим в виде суммы нескольких более простых систем. Информацию о каждой из получившихся систем сохраним отдельно (первый способ хранения коэффициентов $a_{i,j,k}$).

Процесс вычисления линейного полинома разделим на умножения и сложения. Информацию о выполнении сложений и умножений будем сохраним отдельно (второй способ хранения коэффициентов $a_{i,j,k}$).

Способы хранения

Обозначим

$$\tilde{\mathbf{x}}_i = \begin{pmatrix} 1 \\ x_{i,1} \\ \ldots \\ x_{i,n1} \end{pmatrix};$$

$$A_i = \begin{pmatrix} a_{i,1,0} & a_{i,1,1} & \ldots & a_{i,1,ni} \\ \ldots & \ldots & \ldots & \ldots \\ a_{i,mi,0} & a_{i,mi,1} & \ldots & a_{i,mi,ni} \end{pmatrix}; \ \mathbf{y}_i = \begin{pmatrix} y_{i,1} \\ \ldots \\ y_{i,mi} \end{pmatrix}.$$

Тогда (1) преобразуется в алгоритм.

Для $i = 1, \ldots, s$ выполняются вычисления по формулам:

$$\mathbf{y}_i = A_i \tilde{\mathbf{x}}_i;$$

$$x_{i+1,t} = f_{i,t}\left(y_{i,1}, y_{i,2}, \ldots, y_{i,mi}\right), \ t = 1, 2, \ldots, n_{i+1}.$$

Первый способ хранения. Он напоминает традиционный способ хранения разреженных матриц в формате <коэффициент, номер строки, номер столбца>, но за счёт округления появляется много одинаковых коэффициентов, информация о которых содержится отдельно в формате <номер строки, номер столбца>.

Упорядочим все значения ненулевых коэффициенты $a_{i,j,k}$ во всех формулах (1) по убыванию количества присутствий значений данных коэффициентов в функциях $y_{i,j}$, получим список чисел $b_1, b_2, \ldots, b_l$. Рассмотрим часть этого списка $b_1, b_2, \ldots, b_r$.

Для каждой матрицы A_i сохраним:

● r списков пар номеров строк и столбцов <j, k>, на пересечении которых стоят коэффициенты, равные

$$b_k,\ k = 1, \ldots, r;$$

● список троек <$a_{i,j,k}$, j, k> — ненулевой коэффициент, пары номеров строки и столбца, на пересечении которых стоит коэффициент $a_{i,j,k}$.

Пример 2. Пусть

$$y_{1,1} = 1x_{1,1} + 1x_{1,5} - x_{1,6} + 2x_{1,7};$$
$$y_{1,2} = 1x_{1,1} + 1x_{1,3} - 3x_{1,6} + 1x_{1,7}$$

или

$$\mathbf{y}_i = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & -1 & 2 \\ 0 & 1 & 0 & 1 & 0 & 0 & -3 & 1 \end{pmatrix} \tilde{\mathbf{x}}_i.$$

Коэффициент 1 встречается пять раз, коэффициенты –1, 2, –3 — по одному разу. Получим список чисел $b_1, b_2, \ldots, b_l$:1, –1, 2, 3. Рассмотрим часть этого списка $b_1, b_2, \ldots, b_r$:1.

Список пар номеров строк и столбцов, на пересечении которых стоят коэффициенты, равные 1:

(<1,1>, <1,5>, <2,1>, <2,3>, <2,7>).

Список для коэффициентов, не равных 1:

(<–1,1,6>, <2,1,7>, <–3,2,6>).

Второй способ хранения. Для каждого i построим битовую матрицу:

$$C_i = \left(\left(c_{jk}^{(i)} \right)_{j=1,k=0}^{mi,ni} \right),$$

где

$$c_{jk}^{(i)} = \begin{cases} 0, \text{ если } a_{i,j,k} = 0 \\ 1, \text{ если } a_{i,j,k} \neq 0 \end{cases}, \ j = 1, \ldots, m_i, \ k = 0, \ldots n_i.$$

Для каждой матрицы A_i будем хранить:

● матрицу C_i;

● список ненулевых коэффициентов матрицы A_i.

Продолжение примера 2.

$$C_i = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Список ненулевых коэффициентов матрицы A_i:

(1, 1, –1, 2, 1, 1, –3, 1).

Результаты экспериментов

Проведены эксперименты с наборами данных *MNIST*, *USPS*.

Набор данных MNIST — одинакового размера центрированные изображения рукописных цифр. В обучающей выборке — 60000 изображений, а работа построенной нейронной сети проверялась на контрольной выборке из 10000 изображений.

Использована трёхслойная нейронная сеть с 1000 нейронов на первом слое, 200 нейронов — на втором и 10 нейронами — на третьем. Для первых двух слоёв в качестве функции активации использован гиперболический тангенс, а для третьего — номер максимального из значений, т. е.

$$\begin{aligned} &m_1 = 1000;\ m_2 = 200;\ m_3 = 10; \\ &f_{i,t}\left(y_{i,1}, y_{i,2}, \ldots, y_{i,mi}\right) = \operatorname{th} y_{i,t}, i = 1, 2, t = 1, 2, \ldots, n_i; \\ &f_{3,1}\left(y_{3,1}, y_{3,2}, \ldots, y_{3,10}\right) = \underset{1 \le t \le 10}{\operatorname{argmax}}\, y_{3,t}. \end{aligned} \quad (2)$$

Значение метрики *Accuracy* для нейронной сети, без округлений подсчитанной на контрольной выборке, составило 98,27%, а значение той же метрики на той же выборке для нейронной сети с округлёнными коэффициентами, представленными 10-битовыми целыми числами, — 98,24%.

Для хранения коэффициентов нейронной сети без округления и применения технологий хранения разреженных данных потребовалось 7,53 Мб. Для хранения коэффициентов нейронной сети без округления, но с использованием технологии хранения разреженных матриц потребовалось 1,41 Мб.

Информация об объёме памяти, необходимом для хранения округлённых коэффициентов нейронной сети, представленных 10-битовыми целыми числами, с применением первого способа хранения данных показана на рис. 1.

Второй способ хранения данных хорошо показал себя только для внешней памяти на диске. Для хранения округлённых коэффициентов нейронной сети, представленных 10-битовыми целыми числами, потребовалось 0,25 Мб.

Набор данных USPS — одинакового размера изображения рукописных цифр. В обучающей выборке 7291 изображений, а работа построенной нейронной сети проверена на контрольной выборке из 2007 изображений.

Использована нейронная сеть, аналогичная (2), с $m_1 = 254$; $m_2 = 100$; $m_3 = 10$.

Значение метрики *Accuracy* для нейронной сети без округлений, подсчитанной на контрольной выборке,

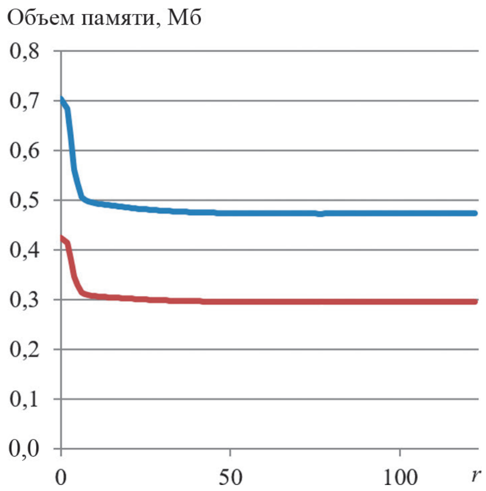

Рис. 1. Объём памяти для хранения коэффициентов для *MNIST*: — в памяти; — на диске

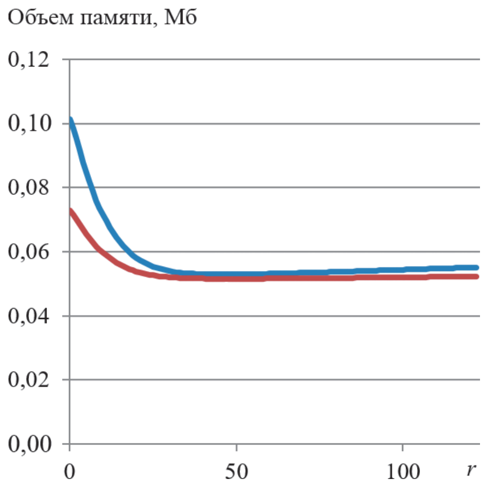

Рис. 2. Объём памяти для хранения коэффициентов для *USPS*: — в памяти; — на диске

составило 94,22%, а значение той же метрики на той же выборке для нейронной сети с округлёнными коэффициентами, представленными 8-битовыми целыми числами, — 94,32%.

Для хранения коэффициентов нейронной сети без округления и применения технологий хранения разреженных данных потребовалось 0,7 Мб. Для хранения коэффициентов нейронной сети без округления, но с использованием технологии хранения разреженных матриц — 0,25 Мб.

Информация об объёме памяти, потребовавшемся для хранения округлённых коэффициентов нейронной сети, представленных 8-битовыми целыми числами, с применением первого способа хранения данных продемонстрирована на рис. 2.

Второй способ хранения данных хорошо показал себя только для внешней памяти на диске. Для хранения округлённых коэффициентов нейронной сети, представленных 8-битовыми целыми числами, потребовалось 0,033 Мб.

Заключение

Показаны способы сокращения требуемого объёма памяти вычислительной системы при классификации с помощью полносвязной нейронной сети с целыми коэффициентами. Объём памяти сокращен за счёт уменьшения избыточности в промежуточных данных.

Результаты работы могут быть применены при использовании полносвязных нейронных сетей для решения различных задач распознавания в условиях аппаратных ограничений.

Литература

1. **Anguita D., Ghio A., Pischiutta S., Ridella S.** A Support Vector Machine with Integer Parameters // Neurocomputing. 2008. V. 72(1—3). Pp. 480—489.

2. **Tschiatschek S., Paul K., Pernkopf F.** Integer Bayesian Network Classifiers // Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Sci. 2014. V. 8726. Pp. 209—224.

3. **Han S., Mao H., Dally W.J.** Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding // Proc. Intern. Conf. Learning Representations. 2016. Pp. 1—14.

4. **Liu B., Wang M., Foroosh H., Tappen M., Penksy M.** Sparse Convolutional Neural Networks // IEEE Conf. Computer Vision and Pattern Recognition. 2015. Pp. 806—814.

5. **Соловьев Р.А., Кустов А.Г., Тельпухов Д.В., Рухлов В.С.** Прототипирование высокоскоростной нейронной сети в ПЛИС для классификации изображений видеопотока // Cloud of Sci. 2018. № 4. С. 680—703.

References

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

2. **Tschiatschek S., Paul K., Pernkopf F.** Integer Bayesian Network Classifiers. Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Sci. 2014;8726:209—224.

3. **Han S., Mao H., Dally W.J.** Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. Proc. Intern. Conf. Learning Representations. 2016:1—14.

4. **Liu B., Wang M., Foroosh H., Tappen M., Penksy M.** Sparse Convolutional Neural Networks. IEEE Conf. Computer Vision and Pattern Recognition. 2015:806—814.

5. **Solov'ev R.A., Kustov A.G., Tel'pukhov D.V., Rukhlov V.S.** Prototipirovanie Vysokoskorostnoy Neyronnoy Seti v PLIS dlya Klassifikatsii Izobrazheniy Videopotoka. Cloud of Sci. 2018;4:680—703. (in Russian).

6. **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.

7. **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.

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

9. **Мамонтов А.И.** О повышении эффективности вычислений при классификации изображений // Вестник МЭИ. 2019. № 5. С. 129—134.

10. **Мамонтов А.И.** О способах экономии памяти при решении задач классификации линейным методом опорных векторов // Вестник МЭИ. 2020. № 4. С. 129—135.

11. **Gorban A.N., Wunsch II D.C.** The General Approximation Theorem // Proc. Intern. Joint Conf. Neural Networks. 1998. Pp. 1271—1274.

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

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

14. **Anguita D., Pischiutta S., Ridella S., Sterpi D.** Feed-forward SVM without Multipliers // IEEE Trans. Neural Networks. 2006. V. 17. Pp. 1328—1331.

15. **Anguita D., Boni A., Ridella S.** A Digital Architecture for Support Vector Machines: Theory, Algorithm and FPGA Implementation // IEEE Trans. Neural Networks. 2003. V. 14. Pp. 993—1009.

16. **Anguita D., Bozza G.** The effects of quantization on support vector machines with Gaussian kernel // IEEE Intern. Joint Conf. Neural Networks. 2005. V. 2. Pp. 681—684.

17. **Lesser B., Mücke M., Gansterer W.N.** Effects of Reduced Precision on Floating-point SVM Classification Accuracy // Procedia Computer Sci. 2011. V. 4. Pp. 508—517.

18. **Tschiatschek S., Pernkopf F.** On Bayesian Network Classifiers with Reduced Precision Parameters // IEEE Trans. Pattern Analysis and Machine Intelligence. 2015. V. 37(4). Pp. 774—785.

19. **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. 2008. V. 276. Pp. 427—431.

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

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

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

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

9. **Mamontov A.I.** O Povyshenii Effektivnosti Vychisleniy pri Klassifikatsii Izobrazheniy. Vestnik MEI. 2019;5:129—134. (in Russian).

10. **Mamontov A.I.** O Sposobakh Ekonomii Pamyati pri Reshenii Zadach Klassifikatsii Lineynym Metodom Opornykh Vektorov. Vestnik MEI. 2020;4:129—135. (in Russian).

11. **Gorban A.N., Wunsch II D.C.** The General Approximation Theorem. Proc. Intern. Joint Conf. Neural Networks. 1998:1271—1274.

12. **Gorban' A.N.** Obobshchennaya Approksimatsionnaya Teorema i Vychislitel'nye Vozmozhnosti Neyronnykh Setey. Sibirskiy Zhurnal Vychislitel'noy Matematiki. 1998;1;1:11—24. (in Russian).

13. **Polovnikov V.S.** O Nelineynykh Kharakteristikakh Neyronnykh Skhem v Proizvol'nykh Bazisakh. Intellektual'nye Sistemy. 2013;17;1—4:87—90. (in Russian).

14. **Anguita D., Pischiutta S., Ridella S., Sterpi D.** Feed-forward SVM without Multipliers. IEEE Trans. Neural Networks. 2006;17:1328—1331.

15. **Anguita D., Boni A., Ridella S.** A Digital Architecture for Support Vector Machines: Theory, Algorithm and FPGA Implementation. IEEE Trans. Neural Networks. 2003;14:993—1009.

16. **Anguita D., Bozza G.** The effects of quantization on support vector machines with Gaussian kernel. IEEE Intern. Joint Conf. Neural Networks. 2005;2:681—684.

17. **Lesser B., Mücke M., Gansterer W.N.** Effects of Reduced Precision on Floating-point SVM Classification Accuracy. Procedia Computer Sci. 2011;4:508—517.

18. **Tschiatschek S., Pernkopf F.** On Bayesian Network Classifiers with Reduced Precision Parameters. IEEE Trans. Pattern Analysis and Machine Intelligence. 2015;37(4):774—785.

19. **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. 2008;276:427—431.

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

Сведения об авторе:

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

Information about author:

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

Работа выполнена при поддержке: РФФИ (проект № 19-01-00294)
The work is executed at support: RFBR (Project No. 19-01-00294)

Статья поступила в редакцию: *12.11.2020*
The article received to the editor: *12.11.2020*