

Раздел I. Алгоритмы обработки информации

УДК 004.89/.85+004.93:656.8

DOI 10.23683/2311-3103-2019-3-6-19

А.О. Пьявченко, А.В. Ильченко

ВОПРОСЫ ПОСТРОЕНИЯ НЕЙРОСЕТЕВОГО КЛАСТЕРИЗАТОРА ДЛЯ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ПОЧТОВОГО ИНДЕКСА

Рассматриваются актуальные вопросы разработки нейросетевого кластеризатора, предназначенного для распознавания рукописного кода почтового индекса корреспонденции, задаваемого с использованием упрощенного кодового штампа. В качестве нейросетевого базиса кластеризатора использована структурированная сверточная нейронная сеть, обеспечивающая формирование и кластеризацию набора классифицирующих признаков, характеризующих символы кода индекса. Разработана программная модель нейросетевого кластеризатора, обеспечивающая посимвольное распознавание сканированного и соответствующим образом предварительно обработанного изображения почтового индекса. При разработке кластеризатора использовался пакет Deep Learning ToolBox программной среды MatLab версии 18b. Для обучения нейросетевого кластеризатора применялся цифровой набор образцов рукописных арабских цифр MNIST. В статье представлены результаты выполненных экспериментов, из которых следует, что обученная программная модель нейросетевого кластеризатора обеспечивает ошибку распознавания произвольной рукописной цифры из состава проверочной выборки на уровне 0,72 %. Преимуществами предлагаемого подхода являются: отказ от применяемого в настоящее время кодового штампа начертания индекса путем замены его штампом упрощенного представления и формально неограниченной длины; возможность использования в составе индекса рукописных символьных обозначений, принятых национальными почтовыми системами при условии соответствующего предварительного обучения нейросетевого кластеризатора; возможность применения предлагаемого подхода на различных этапах обработки корреспонденции, начиная от ее приема у граждан и заканчивая автоматами сортировки в составе крупных сортировочных центров.

Кодовый штамп; почтовый индекс корреспонденции; предварительная обработка; сверточные нейронные сети; обучение с учителем; нейросетевой кластеризатор; моделирование; пакет MATLAB.

А.О. Pyavchenko, A.V. Pchenko

ISSUES OF NEURAL NETWORK CLUSTERIZER CONSTRUCTION FOR ZIP CODE AUTOMATIC RECOGNITION

The article discusses topical issues of a neural network clusterizer development designed to recognize handwritten ZIP code for correspondence, specified using a simplified code stamp. A structured convolutional neural network is used as a neural network basis for clustering. It provides the formation and clustering of classifying features set that characterize the index code characters. A software model of a neural network clusterizer has been developed that provides character-by-character recognition of the scanned and appropriately preprocessed zip code image. In developing the clusterizer we use the MatLab version 18b Deep Learning ToolBox package. A digital set of MNIST handwritten Arabic digits was used to train the neural network clusterizer. The article presents the experiments results, from which it follows that the trained software model of a neural network clusterizer provides an error recognition of an arbitrary handwritten digit from the validation set at the level of 0.72 %. The advantages of the proposed

approach are: the rejection of the currently used code stamp by replacing it with a simplified representation and a formally unlimited length; the possibility of using as part of an index handwritten symbol adopted by the national postal systems, subject to appropriate prior training neural network clusters; the possibility of applying the proposed approach at various stages of processing correspondence, ranging from its reception from citizens and ending with sorting machines as part of large sorting centers.

Code stamp; postal index for correspondence; preprocessing; convolution neural networks; supervised learning; neural network clusterizer; modeling; MATLAB package.

Введение и постановка задачи. В современных условиях постоянно развивающегося технического прогресса, с помощью искусственных нейронных сетей (ИНС) решаются такие различные сложные научно-технические задачи, как кластеризация наборов обрабатываемых данных, распознавание образов на цифровых изображениях, прогнозирование развития ситуаций, управление поведением подвижных объектов и др. При этом следует отметить, что ИНС обладают по сравнению с классическими методами рядом таких преимуществ, как:

- ♦ возможность нахождения вероятностного решения указанных выше функциональных задач с требуемым качеством на предварительном подготовительном этапе, называемом этапом обучения, в условиях относительной неизменности значений ключевых признаков, характеризующих типовую окружающую обстановку (ситуацию), подвергаемую анализу;

- ♦ возможность сохранения ранее достигнутых на этапе обучения результатов в условиях анализа реальной окружающей обстановки (ситуации) при относительной неизменности значений характеризующих ее ключевых признаков. Отсюда и относительная устойчивость ИНС к искажениям исходных данных;

- ♦ наличие структурного параллелизма и масштабируемости производимых ИНС вычислительных операций;

- ♦ ну и, наконец, способность ряда ИНС к самообучению, включая формирование формальных обобщений с целью повышения точности получения значения целевой функции, реализуемой нейросетью в процессе ее жизненного цикла.

Наиболее важной особенностью ИНС является их способность эффективно формировать нелинейные зависимости (классифицирующие признаки), описывающие наборы данных, подвергаемые анализу.

Обозначившиеся в настоящее время успехи в области распознавания рукописного текста, полученные с применением ИНС [10, 13, 15], позволяют по-новому взглянуть на решение задачи архитектурного построения устройств автоматической сортировки почтовых отправлений [13, 15], наделив такие устройства элементами искусственного интеллекта. Настоящая статья посвящена оценке возможности применения нейросетевого подхода к решению задачи автоматического распознавания кода почтового индекса корреспонденции (ПИК) применительно к автоматической сортировке почтовых отправлений. При этом ожидается, что благодаря внедрению ИНС в состав архитектуры автоматов сортировки нового поколения, вероятность правильного распознавания ПИК может быть повышена по сравнению с существующим уровнем при расширении вариантов описания кодового представления индекса.

Почтовый индекс как адрес получателя корреспонденции и вопросы его распознавания. Как известно [14, 15, 18], для идентификации почтовых отправлений (корреспонденции) применяют международные почтовые идентификаторы, включая ПИК. В частности, последний применяется для кодирования почтового отделения получателя и состоит из цифр и/или букв. ПИК используется при автоматизированной сортировке простых и заказных писем, бандеролей, прочей корреспонденции в сортировочных центрах различного уровня. Первый в России и круп-

нейший в Восточной Европе автоматизированный сортировочный центр введен в эксплуатацию в 2009 году на базе итальянского оборудования [18]. В настоящее время почтовый индекс используется в адресных системах ста девяносто двух стран, являющихся членами Всемирного почтового союза, включая Россию [14, 18].

Согласно Федеральному закону Российской Федерации «О почтовой связи» (в редакции от 22.08.2004, № 122-ФЗ), почтовый индекс – «условное *цифровое* обозначение почтового адреса, присваиваемое объекту почтовой связи». В России принята 6-значная система ХХХУУУ, где ХХХ – код города, а УУУ – номер почтового отделения. Исключениями из этой системы являются Москва и Санкт-Петербург. Эти мегаполисы имеют несколько кодов города [18].

Для упрощения распознавания индекса автоматизированным способом в ряде стран, в том числе и в России, применяется специальный шаблон – *кодовый штамп* (рис. 1), правила размещения и заполнения которого на лицевой стороне почтового отправления регламентируются ГОСТ Р 51506-99 (Конверты почтовые. Технические требования. Методы контроля).

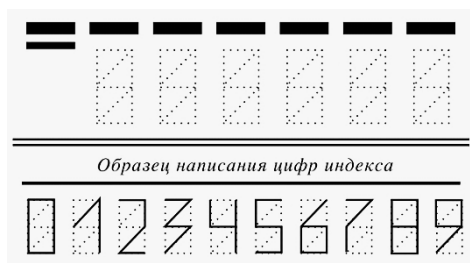


Рис. 1. Индексная сетка и образец ввода цифр ПИК в кодовом штампе России [18]

Главной особенностью кодового штампа является его ручное заполнение человеком-отправителем.

Задача распознавания рукописных цифр ПИК фактически является одной из классических задач распознавания образов [7, 10, 13, 15] и имеет значительную практическую историю [12]. Так, одним из первых практических применений методов распознавания рукописных цифр стало создание системы чтения ZIP-кодов почты США [15]. Используемые в настоящее время в автоматизированных сортировочных центрах автоматы сортировки почтовых отправлений, как правило, оборудованы оптико-электрическими устройствами обнаружения почтового индекса и используют технологию оптического распознавания символов (англ. Optical Character Recognition – OCR) [17].

Анализ существующих систем распознавания рукописного текста по его изображению, полученному в результате сканирования, с учетом поставленной задачи распознавания почтового индекса, заданного рукописным способом, позволяет выделить следующие основные этапы распознавания и перевода сканированной информации в электронный вид для дальнейшего использования и обработки:

1. Оптический поиск реперных меток на лицевой части почтового отправления, вертикально-горизонтальное выравнивание всего изображения относительно них. Сканирование области изображения, содержащего ПИК.
2. Предварительная обработка исходного изображения, включая его фильтрацию и повышение контрастности.
3. Сегментация изображения ПИК, подразумевающая поиск и выделение рукописного текста (цифр) на изображении, разделение текста на составные части (элементы).

4. Поиск и извлечение устойчивых признаков каждого из элементов изображения, выделенных на этапе сегментации.

5. Классификация каждого из элементов изображения, выделенных на этапе сегментации, с использованием сформированной на предыдущем этапе совокупности устойчивых признаков.

6. Окончательная обработка результатов для построения итоговой символьной (кодовой) последовательности по результатам классификации.

Существующие правила формирования почтового индекса имеют определенные преимущества. Так, в качестве устойчивых признаков для распознавания символов ПИК в автоматах сортировки почтовых отправлений применяется метрика Хэмминга [17]. В процессе автоматического распознавания рукописно заполненного кодового штампа выполняется поиск метрики очередного обрабатываемого символа (заполненного знакоместа) в сравнении со всеми эталонными шаблонами. Тот эталонный шаблон, которому соответствует минимальное значение метрики, меньшее или равное некоторому априори заданному пороговому значению, и есть результат поиска. Поиск считается безрезультатным, если найденное минимальное значение метрики выше этого порогового значения.

Следует отметить, что начертание цифр, регламентируемое шаблоном, максимизирует расстояние Хэмминга между ними, сводя вероятность ошибочного распознавания к минимуму (типовое значение $p \leq 0.3$, где p – вероятность ошибки в процессе распознавания каждого отрезка из состава кусочно-линейного изображения цифры на кодовом шаблоне). При этом значение вероятности ошибочного распознавания одной цифры вместо другой равно $p^k \cdot (1 - p)^9$, где k – расстояние между кодировками цифр, а 9 – общее количество отрезков в шаблоне цифры [17]. Следовательно, вероятность ошибки, например, при распознавании цифр 2 и 5 равна $p^5 \cdot (1 - p)^4$, при распознавании цифр 8 и 0: $p \cdot (1 - p)^8$. Таким образом, человеку-отправителю необходимо на лицевой части почтового отправления указать почтовый индекс строго согласно шаблону. Причем, как уже отмечалось выше, делается это вручную рукописным способом в соответствии с примерами на рис. 1.

Несомненным достоинством рассматриваемого подхода является его простота, однако в случае прорисовки цифр со значительными от шаблона отклонениями (при изображении требуемой цифры какие-либо линии не проведены по точкам кодового штампа), либо наличия существенных помарок в изображении цифр, автомат сортировки перенаправит почтовое отправление или на ручную сортировку, или иному адресату при ошибочной интерпретации полученного в результате сканирования числового кода. Как следствие, время доставки корреспонденции адресату может многократно возрасти, что крайне нежелательно в условиях существующей конкуренции почтовых служб.

Также следует отметить, что существующие правила шаблонного начертания ПИК блокируют возможность применения для его кодирования всего разнообразия символов из состава используемого алфавита. Например, по правилам шаблона (рис. 1) могут быть легко начертаны такие символы латинского алфавита, как A, C, E, F, G, H, I, J, L, P, U, а вот символы O, S имеют совпадающее с цифрами представление – соответственно, 0 и 5; символы Q, R, K, T, X, W, M и ряд других не имеют адекватного представления. Отсюда и существующие ограничения на применяемую кодировку.

Сверточные нейронные сети как средство распознавания почтового индекса. Одной из разновидностью ИНС прямого распространения являются сверточные нейронные сети (СНС) (англ. feed-forward). Как правило, СНС используются для эффективного распознавания образов, присутствующих в кадре (на изображении), причем такие сети предварительно должны быть обучены на выполне-

ние необходимой целевой функции [4, 7, 10, 15]. При этом при использовании СНС необязательно производить устранение шумов и прочих дефектов на изображении. Дело в том, что многие фильтры обработки изображения используют операцию свёртки (например, медианная фильтрация). В СНС ключевой операцией как раз и является операция свёртки (1).

$$(I \times K)_{x,y} = \sum_{i=1}^h \sum_{j=1}^w \sum_{k=1}^d K_{ijk} \times I_{x+i-1,y+j-1,k}, \quad (1)$$

где I – двумерное изображение, K – небольшая матрица размерности $h \times w$ (так называемое ядро свертки – kernel), построенная таким образом, что графически кодирует какой-либо признак, $I \times K$ – свернутое изображение (карта признаков).

СНС имеет типовую многослойную однонаправленную структуру и, согласно рис. 2, содержит входной слой (англ. **INPUT**), свёрточный слой (англ. **CONVolution**), слой пакетной нормализации (англ. batch **NORM**alization), функцию активации (англ. **ACTIV**ation function), слой подвыборки (англ. **POOLING** или sub-sampling), полносвязный слой (англ. **DENSE**, чаще fully connected) и функцию потерь (англ. **LOSS** function) [2, 10, 20].

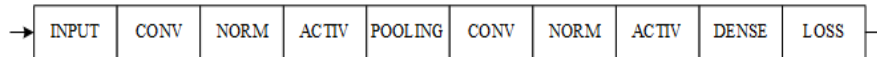


Рис. 2. Типовая структура свёрточной нейронной сети

С учетом результатов исследований [10, 13, 15], именно многослойность СНС позволит при решении задачи распознавания рукописных цифр обеспечить выделение наибольшего количества устойчивых признаков, присущих разным цифрам в условиях высокой вариативности их начертания от руки.

Согласно теории искусственных нейронных сетей их обучение – это процесс определения числовых значений весовых коэффициентов синаптических связей между нейронами, обеспечивающих разбиение пространства признаков на заданное количество кластеров таким образом, чтобы каждому кластеру соответствовали признаки определенного класса. Вычисление весовых коэффициентов синаптических связей осуществляется, как правило, как решение оптимизационной задачи модифицированным методом градиентного спуска [1, 4, 5, 10, 11]. При этом, суть обучения СНС заключается в автоматизированной настройке значений элементов матрицы K таким образом (см. формулу (1)), чтобы заданному на входе образу однозначно соответствовало бы некоторое нелинейное сигнальное распределение значений на выходах сети. Благодаря чему СНС усиливает необходимую информационную составляющую и отбрасывает (подавляет) избыточную [4, 7, 10, 11].

На наш взгляд применение СНС в сортировочных автоматах позволит автоматически распознавать не только предписанное начертание цифр, но и прочие их начертания, а также начертания символов, наборы которых регламентируются национальными почтовыми системами различных стран при условии, что используемая в составе автомата нейронная сеть обучена распознавать такие изображения.

Требования к средствам предварительной обработки и обобщенная структура нейросетевого кластеризатора. Как показывают исследования [10, 13, 15], при одном и том же разрешении на качество работы СНС оказывают сильное влияние цвет, масштаб и аффинные искажения обрабатываемых объектов на изображениях. В целях снижения такого влияния предлагается произвести предварительную обработку исходного изображения, включающую следующую последовательность действий:

1. Поиск реперных меток на изображении и его горизонтально-вертикальное выравнивание
2. Приведение изображения к градациям серого с дополнительной фильтрацией с целью повышения контрастности изображения и отделения текста, в нашем случае, кода ПИК, от используемого фона.
3. Нормализация аффинных искажений.
4. Масштабирование изображения в соответствии с параметрами входного слоя СНС.

Приведение исходного цветного изображения I (RGB-скана) к градациям серого осуществляется по стандартной методике [8], использующей в качестве расчетной формулу вида:

$$J(i, j) = 0.2989R(i, j) + 0.5870G(i, j) + 0.1140B(i, j),$$

где $J(i, j)$ – вещественное значение уровня яркости (интенсивности) каждого пикселя $P(i, j)$ изображения I ;

$R(i, j)$, $G(i, j)$, $B(i, j)$ – уровни яркости соответственно красного, зелёного и синего цветовых каналов для того же пикселя.

Параметрическая нормализация входных данных является важной и обязательной процедурой этапа предобработки. В нашем случае, применение такой процедуры позволяет повысить производительность и стабильность при обучении СНС, а также стабилизировать ее работу в целом [3]. Причем сюда следует отнести как нормализацию возможных аффинных искажений каждого из рукописных символов ПИК, так и их масштабирование – приведение к некоторому единому размеру, принятому за эталон при обработке изображений символов индекса.

На рис. 3,а приведено распределение интенсивностей пикселей на изображении размером 28x28 (пикселей), соответствующее цифре 8. Распределение получено до применения к изображению операций нормализации. При этом использованы следующие обозначения: min matrix value – минимальное значение интенсивности пикселя, max matrix value – максимальное значение, avg matrix value – математическое ожидание интенсивностей, var matrix value – значение оценки их дисперсии. При этом на рисунке на оси абсцисс отмечены интенсивность, на оси ординат – количество пикселей, имеющих соответствующие значения интенсивности.

Одним из наиболее распространенных методов предобработки является нормализация изображения посредством приведения вещественного значения уровня яркости его пикселей к диапазону 0..1. Как видно из рис. 3,б, именно такая нормализация позволяет различить по масштабу изображения символов привести к единому размеру без потери качества самого изображения. Однако при использовании СНС с позиции стабильности и увеличения скорости сходимости алгоритма обучения сети более эффективным является метод нормализации изображения посредством центрирования данных к 0 (англ. zero-centering) [3].

где μ – математическое ожидание, x_i – значение i -го пикселя, σ^2 – оценка дисперсии, ϵ – константа для вычислительной устойчивости (предотвращает деление на ноль).

Из формул (2) следует, что данный метод нормализации обеспечивает центрирование исходного изображения, приводя начальное распределение интенсивностей к единичной дисперсии (var) и нулевому математическому ожиданию (avg), что подтверждается моделированием (см. рис. 3,в).

Предложенная выше последовательность действий при предварительной обработке изображений позволяет отказаться от цифрового кодового штампа в том виде, в котором он применяется в настоящее время, оставив по аналогии с существующим подходом только реперные метки и регламентацию области рукописного ввода символов ПИК с типовым межсимвольным интервалом так, как это показано на рис. 4,а.

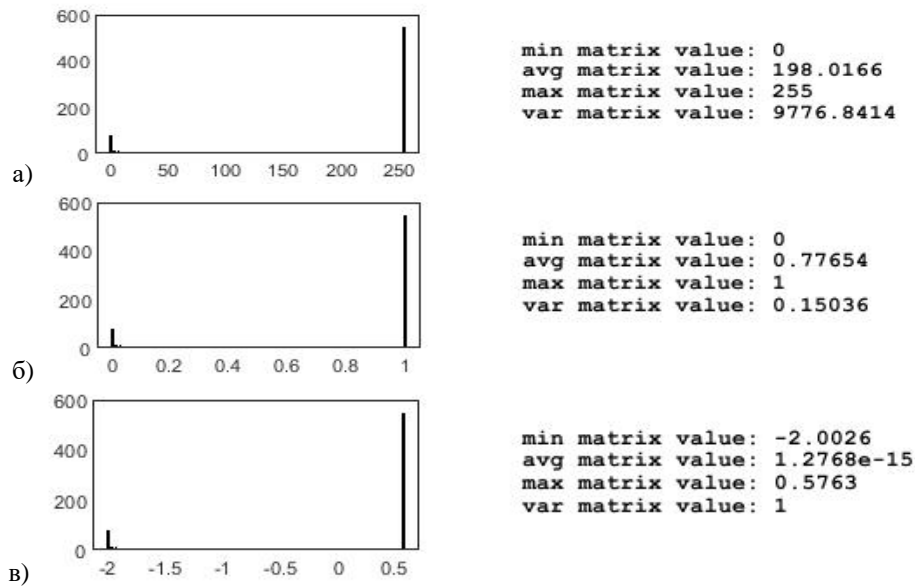


Рис. 3. Свойства данных одного изображения в оттенках серого: а) оригинальное изображение без нормализации; б) нормализация приведением к диапазону 0..1; в) нормализация центрированием данных к нулю

В рамках данного метода обработка входного набора данных x выполняется согласно следующим формулам:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i, \quad \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2, \quad \hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}, \quad (2)$$

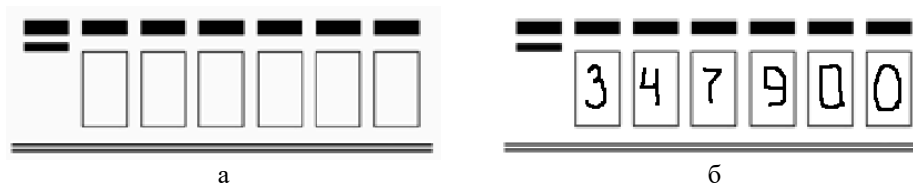


Рис. 4. Предлагаемая индексная сетка при шестизначном ПИК: а) пустой кодовый штамп; б) пример заполнения штампа

Применение такого шаблона, на наш взгляд, обеспечит не только преемственность в используемом оборудовании, но и сделает более разборчивым рукописное написание ПИК, исключит в процессе его обработки необходимость в применении этапа посимвольного разделения рукописного кода индекса, снизит при этом интегральную сложность используемых методов предварительной обработки ПИК. Так, например, наличие реперных меток однозначно определяет область ввода и облегчает решение задачи нормализации аффинных искажений при ручном раздельном написании символов ПИК.

В результате нейросетевой кластеризатор (НСК), как распознаватель кодовой последовательности ПИК, может иметь структуру, соответствующую рис. 5. и включающую устройство приема и сканирования изображения, блоки предвари-

тельной обработки, в том числе, повышения его контрастности, выделения и параметрической нормализации рукописных символов ПИК, модуль СНС, блок окончательной обработки электронного кода ПИК и проверки его на корректность.

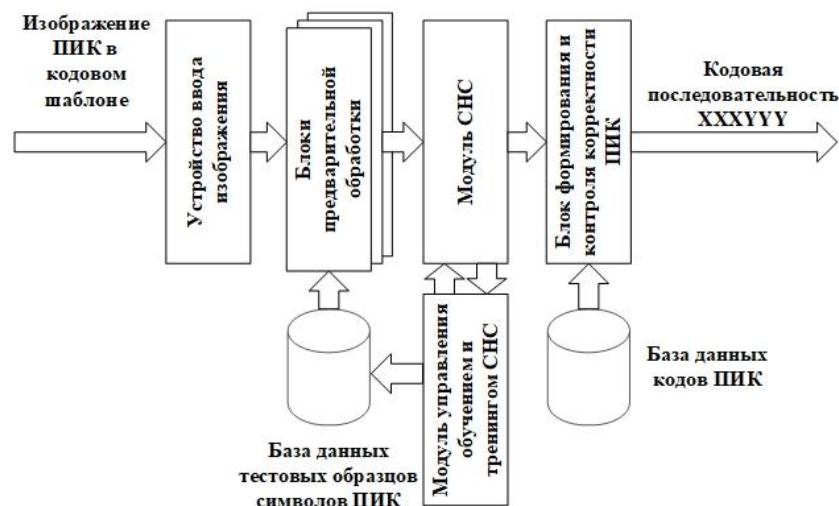


Рис. 5. Предполагаемая структура НСК в составе автомата сортировки

Сам же, поступающий на вход устройства приема и сканирования, код ПИК может быть любым, необязательно цифровым, и записывать его можно будет привычным для человека рукописным способом с некоторыми несущественными ограничениями (см. рис. 4,б).

Постановка эксперимента. Используя средства программного пакета MATLAB, разработаем программную модель НСК, предназначенную для тестирования предложенного принципа распознавания цифр от 0 до 9. При этом в качестве экспериментального набора данных в рамках настоящей работы будем использовать базу данных образцов рукописного написания цифр MNIST [13]. Все образцы приведены к серому полутоновому изображению цифры на белом фоне размером 28x28 пикселей (рис. 6), над ними выполнены операции сглаживания и нормализации [13].

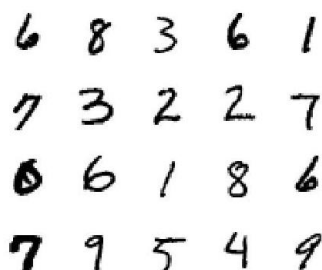


Рис. 6. Двадцать случайных цифр из набора данных MNIST

Перед уточнением структуры НСК необходимо поделить базу данных образцов рукописного написания цифр на обучающую и проверочную выборки. При этом следует найти такое процентное соотношение разбиения, когда получаемые в процессе обучения значения параметров СНС из состава кластеризатора будут иметь малый разброс (дисперсию).

Следуя принципу Парето [16], хорошей отправной точкой при разделении набора данных на обучающую и проверочную выборки является отношение 80/20. В рамках же данной работы было предложено для обучения сети использовать соответствующий набор из базы данных MNIST с отношением обучающей выборки к проверочной как 86/14 из порядка 70000 доступных образцов изображений рукописного написания арабских цифр.

В основу модели НК положим СНС типовой многослойной структуры (см. рис. 2), в которой в качестве слоя подвыборки будем использовать *MaxPool*, в качестве функции потерь *SoftMax*. Указанную сеть необходимо настроить и обучить, максимизировав точность распознавания на проверочной выборке и исключив при этом возникновение эффекта переобучения [4, 5, 9].

Как известно, данная процедура называется «тонкой настройкой» сети (англ. *fine tuning*), а ее параметры, подвергаемые настройке, именуется гиперпараметрами сети [4, 10, 20]. Согласно описанию пакета инструментов Deep Learning Toolbox [6] для MATLAB к гиперпараметрам СНС относят: значение размера ядер свёртки, способ заполнения краевых областей активационных карт (карт признаков), количество этих карт признаков, количество слоев сети, шаг скольжения ядер свертки и субдискретизации, необходимость в нейронах смещения, тип функции активации. Используя метод проб и ошибок, было исследовано влияние значений гиперпараметров, перечисленных выше, на результативность работы СНС изначально выбранной структуры. Полученные результаты сведены в табл. 1, из которой следует, что наилучшую точность удалось достичь на СНС с параметрами, соответствующими строке под номером 5. При этом использовались одни и те же гиперпараметры обучения, подобранные экспериментально.

Таблица 1

Результаты работы СНС с различной структурой на проверочной выборке

№	Кол-во скрытых слоев	Кол-во карт признаков	Размер ядер свёртки	Размер подвыборки	Шаг подвыборки	Функция активации	Ошибка, %
1.	2	8/16	3x3	2x2	2	tanh	1,63
2.	2	8/16	3x3	2x2	2	relu	1,44
3.	2	16/32	3x3	2x2	2	relu	1,15
4.	2	16/32	3x3	2x2	2	tanh	1,38
*5.	3	16/32/64	3x3	2x2	2	relu	0,72

Под гиперпараметрами обучения сети понимают, как правило, режим и метод обучения, скорость обучения и стратегию ее контроля, количество эпох обучения (количество полных циклов обучения по заданному набору данных), частоту проверок результативности и др. При этом, как уже отмечалось выше, задача обучения сети заключается в минимизации общей ошибки сети посредством корректировок весовых коэффициентов нейронной сети по заданному правилу через приближение значения целевой функции к заданной точности [4]. Частота таких корректировок зависит от выбранного режима обучения. В случае стохастического режима корректировка выполняется на каждой итерации работы алгоритма обучения, в пакетном режиме – после просмотра всей обучающей выборки и, наконец, в мини-пакетном режиме (англ. *mini-batch*) просматривается только некоторое подмножество обучающей выборки фиксированного размера (обычно это от 16 до 128 образцов) [9]. Согласно проведенным экспериментам, именно мини-пакетный режим, предусматривающий совмещение стохастического градиентного спуска с пакетным градиентным спуском (метод ADAM [11]), позволяет обеспечить наилучшую сходимость результатов обучения на заданном обучающем наборе за приемлемое машинное время.

В табл. 2 сведены результаты обучения СНС предложенной структуры с целью определения наилучшей точности распознавания на обучающем наборе в зависимости от задаваемых в ходе экспериментов значений гиперпараметров обучения в мини-пакетном режиме.

Процесс обучения СНС с применением значений гиперпараметров обучения, давших наилучшую точность распознавания на обучающей выборке (в нашем случае 99,28 %), показан на рис. 7. Как видно из рисунка, разработанная средствами MATLAB программа тренировала сеть на обучающей выборке в пакетном режиме (при 96 изображениях в пакете) и регулярно вычисляла точность на проверочной выборке (один раз в эпоху при общем количестве эпох, равном 20). Причем за каждую эпоху программа совершила 625 итераций (суммарный объем обучающей выборки был равен 60000 образцов, а размер пакета – 96 образцам). Время полного обучения сети на CPU Intel i5 M480 (8ГБ ОЗУ) без применения CUDA технологии составило 1 час 17 минут 31 секунду.

Таблица 2

Результаты обучения сети с различными параметрами обучения в мини-пакетном режиме

№	Метод обучения	Скорость обучения	Контроль скорости обучения	Кол-во эпох	Размер пакета	Частота проверок	Ошибка, %
1.	sgdm	0,005	постоянная	15	64	937	0,93
2.	sgdm	0,005	постоянная	15	96	625	0,88
3.	sgdm	0,004	постоянная	15	96	625	0,83
4.	sgdm	0,006	постоянная	15	96	625	0,85
5.	sgdm	0,004	0,925/2	15	96	625	0,84
6.	sgdm	0,004	0,925/2	20	96	625	0,81
7.	adam	0,0005	0,925/2	15	64	937	0,91
8.	adam	0,0005	0,925/2	20	96	625	0,77
9.	adam	0,0006	0,925/2	20	96	625	0,82
*10.	adam	0,0004	0,925/2	20	96	625	0,72

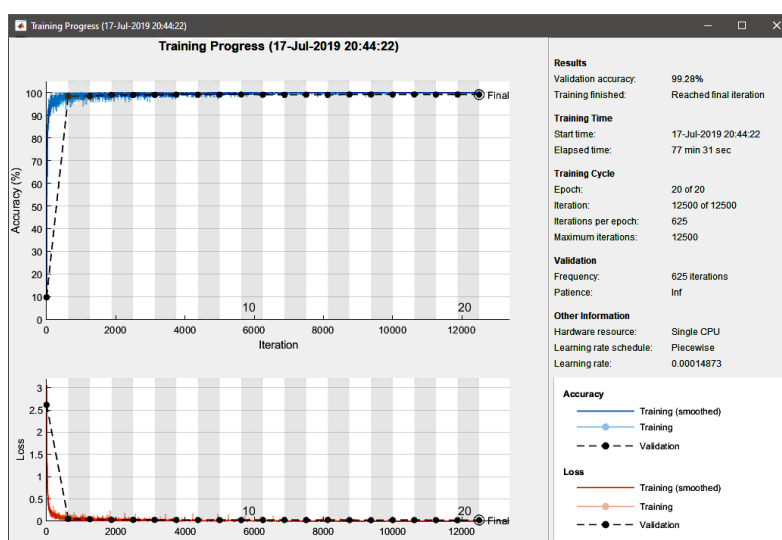
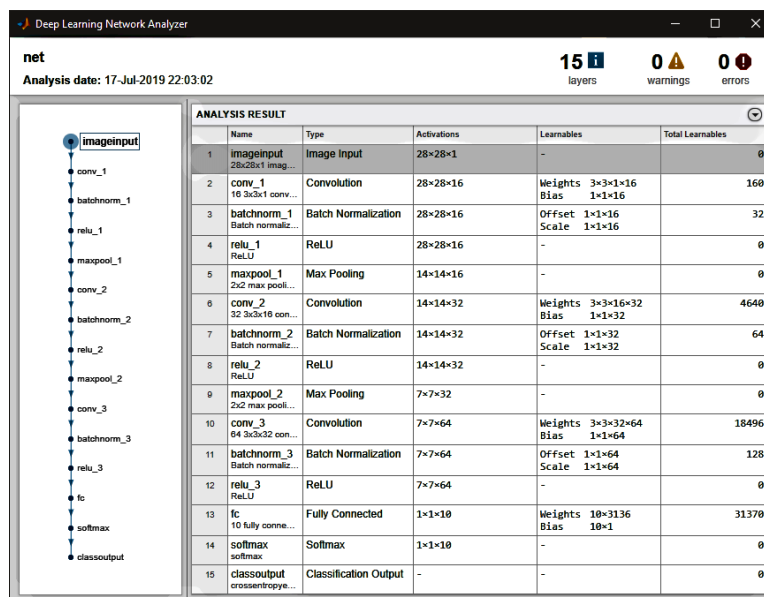


Рис. 7. Результат обучения разработанной сверточной нейронной сети

На рис. 8 приведены значения структурных гиперпараметров модели СНС, полученные в результате ее «тонкой настройки» и выполненного затем системного анализа. В крайней правой колонке указано количество обучаемых параметров (весовые коэффициенты) на каждом слое нейросети.

После завершения построения и обучения разработанной в MATLAB модели НСК, были проведены эксперименты по распознаванию произвольного набора рукописно написанных цифр. Результаты одного из таких экспериментов приведены на рис. 9.



Name	Type	Activations	Learnables	Total Learnables
1 imageinput 28x28x1 imag...	Image Input	28×28×1	-	0
2 conv_1 16 3x3x1 conv...	Convolution	28×28×16	Weights 3×3×1×16 Bias 1×1×16	160
3 batchnorm_1 Batch normaliz...	Batch Normalization	28×28×16	Offset 1×1×16 Scale 1×1×16	32
4 relu_1 ReLU	ReLU	28×28×16	-	0
5 maxpool_1 2x2 max pool...	Max Pooling	14×14×16	-	0
6 conv_2 32 3x3x16 con...	Convolution	14×14×32	Weights 3×3×16×32 Bias 1×1×32	4640
7 batchnorm_2 Batch normaliz...	Batch Normalization	14×14×32	Offset 1×1×32 Scale 1×1×32	64
8 relu_2 ReLU	ReLU	14×14×32	-	0
9 maxpool_2 2x2 max pool...	Max Pooling	7×7×32	-	0
10 conv_3 64 3x3x32 con...	Convolution	7×7×64	Weights 3×3×32×64 Bias 1×1×64	18496
11 batchnorm_3 Batch normaliz...	Batch Normalization	7×7×64	Offset 1×1×64 Scale 1×1×64	128
12 relu_3 ReLU	ReLU	7×7×64	-	0
13 fc 10 fully conne...	Fully Connected	1×1×10	Weights 10×3136 Bias 10×1	31370
14 softmax softmax	Softmax	1×1×10	-	0
15 classoutput crossentropy...	Classification Output	-	-	0

Рис. 8. Результат анализа представленной архитектуры сети

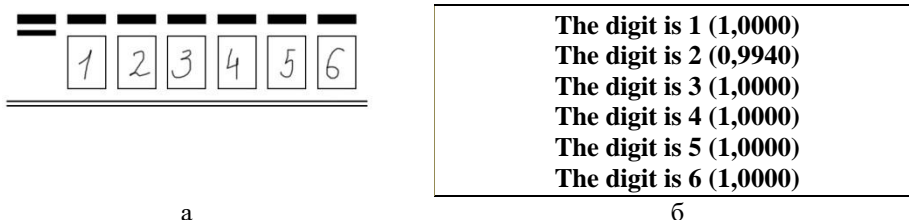


Рис. 9. Ответ СНС посимвольного распознавания рукописного ПИК: а – исходный рукописно заданный код, б – результаты распознавания цифры, в скобках указана вероятность распознавания

Как показали выполненные эксперименты, можно с уверенностью утверждать, что спроектированный НСК посимвольно распознает арабские цифры от 0 до 9, рукописно записанные разборчивым почерком (рис. 9,а), с вероятностью не ниже 0,99 (например, рис. 9,б). Причем НСК смог успешно определить все цифры из предложенной авторами экспериментальной выборки.

Итак, предлагаемый к реализации НСК интеллектуального автомата сортировки принимает решение на основе вероятностной оценки, производимой им автоматически. Аналогичным способом может быть разработан НСК, предназначенный для распознавания наборов символов, смешанных наборов символов и цифр.

Причем архитектура НСК, тип распознаваемых символов целиком зависит от используемого обучающего набора и выбранных структурных гиперпараметров и гиперпараметров обучения.

Выводы. На наш взгляд, преимуществами рассмотренного в статье подхода являются:

- ◆ отказ от применяемого кодового штампа начертания индекса путем замены его штампом упрощенного представления и формально неограниченной длины;
- ◆ переход к рукописному посимвольному представлению ПИК в соответствии с обозначениями, принятыми национальными почтовыми системами при соответствующем предварительном обучении нейросетевого кластеризатора, а также при разборчивости (читабельности) ручного начертания самих символов в шаблоне кодового штампа упрощенного представления;
- ◆ возможность использования предлагаемого подхода на различных этапах обработки корреспонденции, начиная от ее приема у граждан и заканчивая автоматами сортировки в составе крупных сортировочных центров;
- ◆ возможность аппаратной реализации предобученного нейросетевого кластеризатора в базе современных ПЛИС с использованием современных технологий автоматизации HDL-синтеза по аналогии с [19, 21].

В настоящее время предлагаемый подход к построению НСК востребован в области ЖКХ, в банковском деле, в других областях, где после ручного заполнения квитанций, бланков, анкет требуется автоматизированный ввод в ЭВМ заполненных вручную данных с последующим распознаванием содержимого и формированием информационного фрейма с отправкой его потребителю. Тем самым становится возможным снизить уровень ручного труда при компьютерном наборе текста подобных документов за счет внедрения интеллектуальных методов распознавания, отводя оператору информационно-советующую, контролирующую и корректирующую функции.

Работа выполнена в Южном федеральном университете при финансовой поддержке Минобрнауки РФ в рамках реализации проектной части госзадания № 2.3928.2017/4.6.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Convolutional Neural Networks for Visual Recognition. – Backpropagation, Intuitions // Курс лекций CS231n стэнфордского университета (дата обновления: 2019). – URL: <http://cs231n.github.io/optimization-2/> (дата обращения: 17.07.2019).
2. Convolutional Neural Networks for Visual Recognition. – Neural Networks Part 1: Setting up the Architecture // Курс лекций CS231n стэнфордского университета (дата обновления: 2019). – URL: <http://cs231n.github.io/neural-networks-1/> (дата обращения: 17.07.2019).
3. Convolutional Neural Networks for Visual Recognition. – Neural Networks Part 2: Setting up the Data and the Loss // Курс лекций CS231n стэнфордского университета (дата обновления: 2019). – URL: <http://cs231n.github.io/neural-networks-2/> (дата обращения: 17.07.2019).
4. Convolutional Neural Networks for Visual Recognition. – Neural Networks Part 3: Learning and Evaluation // Курс лекций CS231n стэнфордского университета (дата обновления: 2019). – URL: <http://cs231n.github.io/neural-networks-3/> (дата обращения: 17.07.2019).
5. Convolutional Neural Networks for Visual Recognition. – Optimization: Stochastic Gradient Descent // Курс лекций CS231n стэнфордского университета (дата обновления: 2019). – URL: <http://cs231n.github.io/optimization-1/> (дата обращения: 17.07.2019).
6. Deep Learning Toolbox // Официальный сайт компании Mathworks (дата обновления: 2019). – URL: <https://www.mathworks.com/products/deep-learning.html> (дата обращения: 20.07.2019).
7. *Deng Li and Dong Yu*. Deep Learning: Methods and Applications // Foundations and Trends in Signal Processing. – 2013. – Vol. 7, No. 3–4. – P. 197–387.

8. *Gonzalez R.C., Woods R.E.* Digital Image Processing. – 3rd ed. – Prentice-Hall, 2008. – 976 p.
9. *Ioffe, Sergey, and Christian Szegedy.* Batch normalization: Accelerating deep network training by reducing internal covariate shift // arXiv preprint. – 2015. – arXiv:1502.03167.
10. *Schmidhuber J.* Deep Learning in Neural Networks: An Overview // Neural Networks. – January 2015. – Vol. 61. – P. 85-117.
11. *Kingma D., Ba J.* Adam: A method for stochastic optimization // International Conference on Multimedia. – 2014. – P. 675-678.
12. *Mansi Shah, Gordhan B Jethava.* A literature review on hand written character recognition // Indian Streams Research Journal. – March 2013. – Vol. 3, Issue 2.
13. MNIST handwritten digit database // Официальный сайт. – URL: <http://yann.lecun.com/exdb/mnist/> (дата обращения: 17.07.2019).
14. Universal Postal Union. – Universal POST*CODE® DataBase // Официальный сайт почтового союза (дата обновления: 05.2019). – URL: <http://www.upu.int/en/resources/postcodes/universal-postcoder-database.html> (дата обращения: 05.07.2019).
15. *Y. LeCun et al.* Backpropagation Applied to Handwritten Zip Code Recognition // Neural Computation. – Dec. 1989. – No. 1 (4). – P. 541-551.
16. Закон Парето / Авторы Википедии // Википедия (дата обновления: 06.06.2019). – URL: https://ru.wikipedia.org/wiki/Закон_Парето (дата обращения: 05.07.2019).
17. *Акулич И.* Идеальный почтовый индекс // Популярный журнал «Квантик». – 2015. – № 1. – С. 10-14.
18. Почтовый индекс / Авторы Википедии // Википедия (дата обновления: 24.06.2019). – URL: https://ru.wikipedia.org/wiki/Почтовый_индекс (дата обращения: 05.07.2019).
19. *Соловьев Р.А., Кустов А.Г., Рухлов В.С., Щелоков А.Н., Пузырьков Д.В.* Аппаратная реализация свёрточной нейронной сети в ПЛИС на базе вычислений с фиксированной точкой // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 186-187.
20. *Гудфеллоу Я., Бенджио И., Курвилль А.* Глубокое обучение: пер. с англ. А.А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.
21. *Доронченко Ю.И., Коваленко А.Г.* Методика реализации на реконфигурируемых вычислительных системах крупных узлов вычислительных задач в виде библиотечных VHDL-элементов // Известия ЮФУ. Технические науки. – 2016. – № 11 (184). – С. 4-13.

REFERENCE

1. Convolutional Neural Networks for Visual Recognition. Backpropagation, Intuitions, *Kurs leksiy CS231n stendfordskogo universiteta* [Stanford CS231n 2016 Lectures] (Last changes: 2019). Available at: <http://cs231n.github.io/optimization-2/> (accessed: 17 July 2019).
2. Convolutional Neural Networks for Visual Recognition. Neural Networks Part 1: Setting up the Architecture, *Kurs leksiy CS231n stendfordskogo universiteta* [Stanford CS231n 2016 Lectures]. Available at: <http://cs231n.github.io/neural-networks-1/> (accessed: 17 July 2019).
3. Convolutional Neural Networks for Visual Recognition. Neural Networks Part 2: Setting up the Data and the Loss, *Kurs leksiy CS231n stendfordskogo universiteta* [Stanford CS231n 2016 Lectures] (Last changes: 2019). Available at: <http://cs231n.github.io/neural-networks-2/> (accessed: 17 July 2019).
4. Convolutional Neural Networks for Visual Recognition. Neural Networks Part 3: Learning and Evaluation, *Kurs leksiy CS231n stendfordskogo universiteta* [Stanford CS231n 2016 Lectures] (Last changes: 2019). Available at: <http://cs231n.github.io/neural-networks-3/> (accessed: 17 July 2019).
5. Convolutional Neural Networks for Visual Recognition. Optimization: Stochastic Gradient Descent, *Kurs leksiy CS231n stendfordskogo universiteta* [Stanford CS231n 2016 Lectures] (Last changes: 2019). Available at: <http://cs231n.github.io/optimization-1/> (accessed: 17 July 2019).
6. Deep Learning Toolbox, *Ofitsial'nyy sayt kompanii Mathworks* [Official world-wide web site of the Mathworks company] (Last changes: 2019). Available at: <https://www.mathworks.com/products/deep-learning.html> (accessed: 20 July 2019).
7. *Deng Li and Dong Yu.* Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*, 2013, Vol. 7, No. 3–4, pp. 197-387.
8. *Gonzalez R.C., Woods R.E.* Digital Image Processing. 3rd ed. Prentice-Hall, 2008, 976 p.
9. *Ioffe, Sergey, and Christian Szegedy.* Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint*, 2015, arXiv:1502.03167.

10. *Schmidhuber J.* Deep Learning in Neural Networks: An Overview, *Neural Networks*, January 2015, Vol. 61, pp. 85-117.
11. *Kingma D., Ba J.* Adam: A method for stochastic optimization, *International Conference on Multimedia*, 2014, pp. 675-678.
12. *Mansi Shah, Gordhan B Jethava.* A literature review on hand written character recognition, *Indian Streams Research Journal*, March 2013, Vol. 3, Issue 2.
13. MNIST handwritten digit database, *Ofitsial'nyy sayt* [Official web site]. Available at: <http://yann.lecun.com/exdb/mnist/> (accessed: 17 July 2019).
14. Universal Postal Union. – Universal POST*CODE® DataBase, *Ofitsial'nyy sayt pochtoivogo soyuza* [Official postal union web site] (Last changes: 05.2019). Available at: <http://www.upu.int/en/resources/postcodes/universal-postcoder-database.html> (accessed: 05 July 2019).
15. *Y. LeCun et al.* Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, Dec. 1989, No. 1 (4), pp. 541-551.
16. Zakon Pareto [Pareto principle], Wikipedia contributors, *Vikipediya* (Last changes: 06 June 2019). Available at: https://ru.wikipedia.org/wiki/Zakon_Pareto (accessed: 05 July 2019).
17. *Akulich I.* Ideal'nyy pochtoivyy indeks [Perfect zip code], *Populyarnyy zhurnal «Kvantik»* [The popular magazine «Quantic»], 2015, No. 1, pp. 10-14.
18. Pochtoivyy indeks [Postal code], Wikipedia contributors, *Vikipediya* (Last changes: 06 June 2019). Available at: [https://ru.wikipedia.org/wiki/Postal code](https://ru.wikipedia.org/wiki/Postal_code) (accessed: 05 July 2019).
19. *Solov'ev R.A., Kustov A.G., Rukhlov V.S., Shchelokov A.N., Puzyr'kov D.V.* Apparatnaya realizatsiya svertochnoy neyronnoy seti v PLIS na baze vychisleniy s fiksirovannoy tochkoy [Hardware implementation of a convolutional neural network in FPGA based on fixed point calculations], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7 (192), pp.: 186-187.
20. *Gudfellou YA., Bendzhio I., Kurvill' A.* Glubokoe obuchenie [Deep learning]: translation from English A.A. Slinkina. 2nd ed. Moscow: DMK Press, 2018, 652 p.
21. *Doronchenko Yu.I., Kovalenko A.G.* Metodika realizatsii na rekonfiguriruemyykh vychislitel'nykh sistemakh krupnykh uzlov vychislitel'nykh zadach v vide biblioteknykh VHDL-elementov [Methodology of implementation of large nodes of computational problems as library VHDL-units on reconfigurable computer systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 11 (184), pp. 4-13.

Статью рекомендовал к опубликованию д.т.н., профессор А.Ф. Рогачев.

Пьявченко Алексей Олегович – Южный федеральный университет; e-mail: aop61@mail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: +78634371656; кафедра вычислительной техники; к.т.н.; доцент.

Ильченко Артем Владимирович – e-mail: gamerx.ar@gmail.com; тел.: +79081866792; кафедра вычислительной техники; аспирант.

Pyavchenko Aleksey Olegovich – Southern Federal University; e-mail: aop61@mail.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371656; the department of computer engineering; cand. of eng. sc.; associate professor.

Pchenko Artem Vladimirovich – e-mail: gamerx.ar@gmail.com; phone: +79081866792; the department of computer engineering; postgraduate student.