

РАЗРАБОТКА ПРИЛОЖЕНИЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ ЦИФР С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

DEVELOPMENT OF AN APPLICATION FOR RECOGNITION OF HANDWRITTEN NUMBERS USING NEURAL NETWORK TECHNOLOGIES

Т. В. Бучукова^{1,2}, И. В. Лефанова^{1,2}, И. В. Мартинкевич^{1,2}

T. V. Buchukova^{1,2}, I. V. Lefanova^{1,2}, I. V. Martinkevich^{1,2}

¹Белорусский государственный университет, БГУ, г. Минск, Республика Беларусь

²Учреждение образования «Международный государственный экологический институт имени А. Д. Сахарова» Белорусского государственного университета, МГЭИ им. А. Д. Сахарова БГУ),

г. Минск, Республика Беларусь

buchukova.tat@gmail.com

¹Belarusian State University, BSU, Minsk, the Republic of Belarus

²International Sakharov Environmental Institute of Belarusian State University, ISEI BSU, Minsk, the Republic of Belarus

В статье описана разработка приложения распознавания рукописных цифр средствами нейросетевых технологий.

The article describes the development of a handwritten digit recognition application using neural network technologies.

Ключевые слова: Искусственная нейронная сеть, распознавание образов, графический пользовательский интерфейс, сверточные нейронные сети.

Keywords: Artificial neural network, pattern recognition, graphical user interface, convolutional neural networks.

<https://doi.org/10.46646/SAKH-2023-2-291-294>

Постановка задачи. Распознавание рукописных символов является одной из базовых задач компьютерного зрения и дает возможность не только «увидеть» надписи, но и «понять» их смысл. В статье остановимся на решении частной задачи: разработку приложения, принимающего в качестве входных данных изображение любой цифры в виде массива 28х28 пикселей и выдающего на выходе саму «сущность» – цифру от 0 до 9 с определенной долей вероятности.

Так как базовым является массив 28х28 точек, задействовано будет 784 нейрона, содержащие цифры от 0 до 1, чем ближе значение к 1, тем ближе цвет точки к белому. Числа определим как активации нейронов. Данные нейроны образуют входной слой нейронной сети. Выходной слой содержит 10 нейронов для цифр от 0 до 9. В выходном слое каждому из нейронов также соответствует цифра от 0 до 1, отражающая «уверенность» приложения в том, что входное изображение соответствует конкретной цифре. Чем ближе значение конкретного нейрона к 1, тем «увереннее» приложение в том, что пользователь написал конкретную цифру. Также используются средние слои, называемые скрытыми, их число и количество содержащихся в них нейронов определяется исходя из поставленной задачи. В описываемом тестовом примере используется один скрытый слой с 256 нейронами.

Скрытый слой необходим для определения состава каждой цифры (например, цифру «четыре» можно представить в качестве трех определенным образом расположенных и соединенных между собой линий, цифру «девять» – в качестве окружности и линии справа, цифру «восемь» – в качестве двух окружностей, расположенных друг над другом и имеющих одну общую точку).

Переход от скрытого слоя к выходному может, таким образом, представлять собой соответствие набора элементов какой-либо цифре. Вместе с тем для достижения высокой степени точности или распознавания большого набора символов каждый элемент можно разбить на составляющие. Таким образом введенное изображение приводит к активации определенных нейронов первого скрытого слоя, определяющих характерные составляющие элементов цифр, эти нейроны в свою очередь активируют более крупные формы (элементы), и в результате активируется нейрон выходного слоя, ассоциированный определенной цифре.

Для решения поставленной задачи необходимо определить алгоритм обучения искусственной нейронной сети, который будет находить значения весов и смещения. Кроме того, необходим тренировочный набор данных для обучения нейронной сети. Под итерацией будем понимать общее количество эталонных данных для обучения. Эпоха является счетчиком прохождений всего набора эталонных данных, при этом, чем больше значение данного счетчика, тем лучше обучена искусственная нейронная сеть и соответственно тем точнее получаемый приложением результат. Ошибка представляет собой процентную величину, отражающую расхождение ожидаемого

и полученного результата, при этом ошибка формируется каждую эпоху и с увеличением эпохи должна идти на спад. После каждой эпохи необходимо изменять веса для уменьшения ошибки. Процесс обучения искусственной нейронной сети таким образом сводится к уменьшению ошибки.

Материалы и методы исследования. В качестве эталонного набора данных используется MNIST (сокращение от «Modified National Institute of Standards and Technology») – объёмная база данных образцов рукописного написания цифр. Данный набор был сформирован Национальным институтом стандартов и технологий США. В качестве базы были взяты образцы написания цифр из результатов переписи населения. С течением времени к ним добавлялись другие варианты, например, взятые из тестирований студентов и т.п. В эталонном наборе (датасете) содержится 70 тысяч картинок с цифрами от 0 до 9, приведенных к одинаковому виду (черный фон – центрированная белая цифра 20×20 пикселей), имеющему размер 28×28 пикселей. Набор разделен на две неравные части: train (60 тысяч изображений, которые используются для обучения) и test (10 тысяч изображений, которые используются в качестве тестовой выборки для проверки результатов обучения).

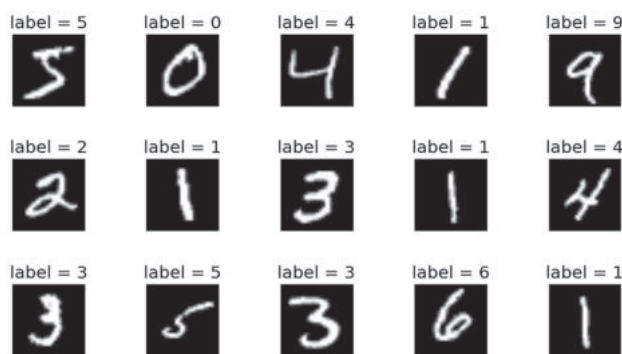


Рисунок 1 – Набор данных MNIST

Непосредственно для разработки приложения использовались язык программирования Python 3.9, библиотеки глубокого обучения TensorFlow и Keras, библиотека для разработки графического пользовательского интерфейса Tkinter. Библиотека Keras содержит в своем составе эталонный набор данных MNIST. Разработка велась в интерактивном блокноте Jupyter Notebook.

Первым шагом является подключение, обработка и преобразование набора данных

```
import keras
from keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('Размерность x_train:', x_train.shape)
print(x_train.shape[0], 'Размер выборки train')
print(x_test.shape[0], 'Размер выборки test')
```

После загрузки, обработки и преобразования обучающей выборки создаем сверточную нейронную сеть, описанную ранее и обучаем модель, используя функцию `model.fit()` из библиотеки Keras, которая принимает тренировочные, валидационные данные, эпохи (epoch) и размер батча (batch). После обучения веса и определение модели сохраняются в файле `mnist.h5`. Далее производится оценка качества обучения модели.

Для обучения было установлено десять эпох, как видно на рисунке 2 при прохождении эпох от 1 до 10, уменьшаются потери на тесте и увеличивается точность обучения на тесте.

Завершающим этапом разработки приложения является разработка графического пользовательского интерфейса: стандартное окно Windows, содержащее поле для рисования цифры (выделено серым), поле для вывода результатов распознавания и двух кнопок: для запуска процесса распознавания и для очистки поля для рисования.

```

(60000, 28, 28) (60000,)
Размерность x_train: (60000, 28, 28, 1)
60000 Размер train
10000 Размер test
Epoch 1/10
469/469 [=====] - 71s 151ms/step - loss: 2.2663 - accuracy: 0.1621 - val_loss: 2.2077 - val_accuracy: 0.4163
Epoch 2/10
469/469 [=====] - 73s 155ms/step - loss: 2.1739 - accuracy: 0.3151 - val_loss: 2.0961 - val_accuracy: 0.6086
Epoch 3/10
469/469 [=====] - 69s 148ms/step - loss: 2.0587 - accuracy: 0.4541 - val_loss: 1.9488 - val_accuracy: 0.7016
Epoch 4/10
469/469 [=====] - 70s 149ms/step - loss: 1.9030 - accuracy: 0.5550 - val_loss: 1.7535 - val_accuracy: 0.7486
Epoch 5/10
469/469 [=====] - 71s 152ms/step - loss: 1.7097 - accuracy: 0.6157 - val_loss: 1.5161 - val_accuracy: 0.7748
Epoch 6/10
469/469 [=====] - 69s 147ms/step - loss: 1.4873 - accuracy: 0.6601 - val_loss: 1.2664 - val_accuracy: 0.7937
Epoch 7/10
469/469 [=====] - 69s 147ms/step - loss: 1.2809 - accuracy: 0.6862 - val_loss: 1.0480 - val_accuracy: 0.8123
Epoch 8/10
469/469 [=====] - 70s 149ms/step - loss: 1.1127 - accuracy: 0.7065 - val_loss: 0.8826 - val_accuracy: 0.8269
Epoch 9/10
469/469 [=====] - 71s 151ms/step - loss: 0.9870 - accuracy: 0.7296 - val_loss: 0.7648 - val_accuracy: 0.8366
Epoch 10/10
469/469 [=====] - 71s 150ms/step - loss: 0.8924 - accuracy: 0.7454 - val_loss: 0.6802 - val_accuracy: 0.8476

```

Рисунок 2 – Процесс обучения модели

```

Модель успешно обучена
Модель сохранена как mnist.h5
Потери на тесте: 0.6801786422729492
Точность на тесте: 0.847599983215332

```

Рисунок 3 – Результат обучения модели

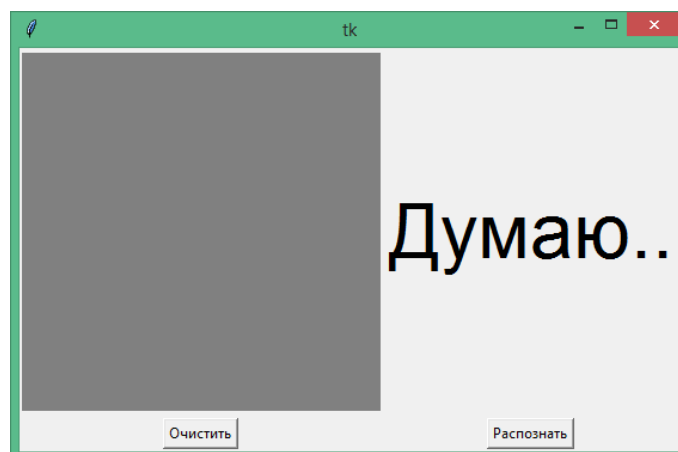


Рисунок 4 – Графический пользовательский интерфейс приложения

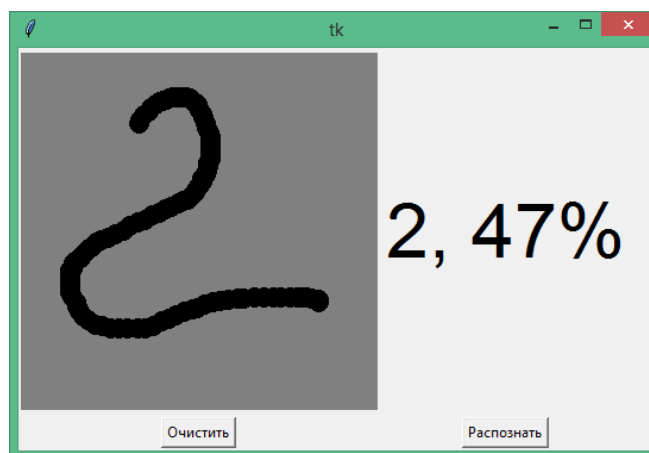


Рисунок 5 – Тестирование работы приложения и правильности обучения модели на примере цифры «2»

ЛИТЕРАТУРА

1. *Bishop, C.M. Neural Networks for Pattern Recognition*/C.M. Bishop – Oxford University Press, 1995. 485 p.
2. *Mc Culloch W.C. H. A Logical Calculus of Ideas Immanent in Nervous Activity.* / W. C. Mc Culloch, W. H. Pitts. // Bulletin of Mathematical Biophysics. – 1943. – Vol. 5. – p. 115–119.

РАЗРАБОТКА МОДИФИЦИРОВАННОЙ МОДЕЛИ SEIR ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИКИ РАСПРОСТРАНЕНИЯ ЭПИДЕМИЙ ИНФЕКЦИОННЫХ ЗАБОЛЕВАНИЙ

DEVELOPMENT OF A MODIFIED SEIR MODEL FOR MODELING THE DYNAMICS OF THE SPREAD OF EPIDEMICS OF INFECTIOUS DISEASES

И. В. Лефанова^{1,2}, Т. В. Смирнова^{1,2}

I. V. Lefanova^{1,2}, T. V. Smirnova^{1,2}

¹Белорусский государственный университет, БГУ, г. Минск, Республика Беларусь

²Учреждение образования «Международный государственный экологический институт имени А. Д. Сахарова» Белорусского государственного университета, МГЭИ им. А. Д. Сахарова БГУ, г. Минск, Республика Беларусь
irina.lefanova@mail.ru

¹Belarusian State University, BSU, Minsk, the Republic of Belarus

²International Sakharov Environmental Institute of Belarusian State University, ISEI BSU, Minsk, the Republic of Belarus

В статье описывается разработка модифицированной модели SEIR для математического и компьютерного моделирования динамики распространения эпидемий инфекционных заболеваний. Передаточные коэффициенты модели рассчитываются на основе клинических данных из открытых источников.

The article describes the development of a modified SEIR model for mathematical and computer modeling of the dynamics of the spread of epidemics of infectious diseases. The transfer coefficients of the model are calculated based on clinical data from open sources.

Ключевые слова: компартментные модели, модель SEIR, эпидемиология, эпидемия, математическое моделирование, компьютерное моделирование

Keywords: compartment models, SEIR model, epidemiology, epidemic, mathematical modeling, computer modeling.

<https://doi.org/10.46646/SAKH-2023-2-294-297>

Компартментные модели основаны на законе сохранения масс и особенностях передачи инфекции. Математическое моделирование в эпидемиологии фактически началось с работы D. Bernoulli в 1760 году, обосновывающей эффективность проведения вакцинации населения против ветряной оспы. Впоследствии появилась математическая модель, основанная на законе баланса масс, которые в итоге дали основу для создания компартментной SIR-модели W.O. Kermack и A.G. McKendrick.

В рамках математического моделирования было построено следующая схема распространения эпидемии:

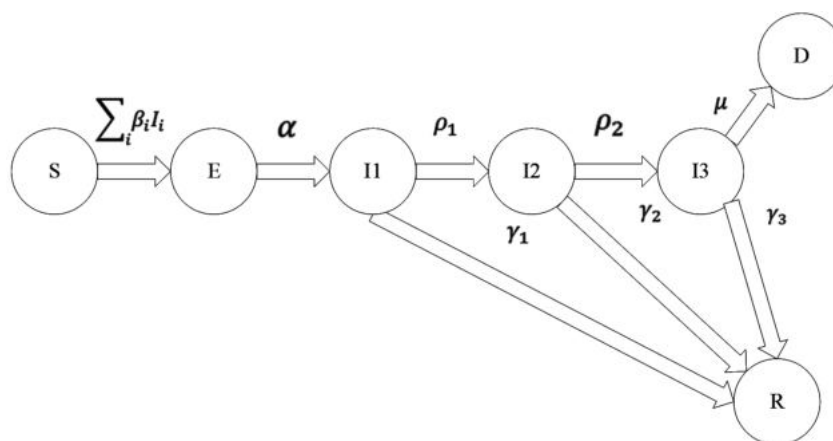


Рисунок 1 – Диаграмма передачи для модифицированной модели SEIR