МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федерального государственного бюджетного образовательного учреждения высшего образования

**«РОССИЙСКИЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ   
Г.В.ПЛЕХАНОВА»**

**Техникум Пермского института (филиала)**

Индивидуальный проект

по дисциплине: «Введение в специальность»

на тему: «**Нейросети. Программа по распознаванию цифр методом нейросетевого моделирования**»

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.О. Фамилия/

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Исполнитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.О. Фамилия/

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г

Пермь, 2021 г.

Оглавление

[**ВВЕДЕНИЕ** 3](#_Toc72835277)

[1 Нейросети 4](#_Toc72835278)

[1.1 Краткая история ИНС 4](#_Toc72835279)

[1.2 Искусственный нейрон 5](#_Toc72835280)

[1.2.1 Перцептрон 8](#_Toc72835281)

[1.2.2 Функция активации 11](#_Toc72835282)

[1.3 Архитектура нейронных сетей 12](#_Toc72835283)

[1.4 Методы обучения ИНС 15](#_Toc72835284)

[2 Распознавание образов 18](#_Toc72835285)

[2.1 Принципы распознавания образов 18](#_Toc72835286)

[2.2 Система распознавания образов 20](#_Toc72835287)

[2.3 Задача классификации образов 22](#_Toc72835288)

[3 Создание программы 24](#_Toc72835289)

[3.1. Распознавание цифр методом нейросетевого моделирования 24](#_Toc72835290)

[3.2 Ход работы и отладка 26](#_Toc72835291)

[3.3 Код программы и примеры результата выполнения 28](#_Toc72835292)

[**ЗАКЛЮЧЕНИЕ** 31](#_Toc72835293)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 32](#_Toc72835294)

# **ВВЕДЕНИЕ**

С момента своего появления искусственные нейронные сети (ИНС) используются как для решения различных прикладных задач, так и для изучения возможных правил функционирования мозга. ИНС успешно применяются в широчайшем спектре приложений, таких как распознавание образов, прогнозирование, выявление зависимостей, сжатие данных, задачи управления и многие другие. Одно из важных направлений практического применения ИНС – задачи распознавания.

В современном мире распознавание образов находит все большее применение в повседневной жизни людей. Методы и алгоритмы теории распознавания широко применяются в медицине (диагностика медицинских снимков), геологии (изучение природных ресурсов Земли), робототехнике (зрение роботов), астрономии, при анализах изображений, идентификации человека, автоматическом проектировании и т.д.

В связи с вышеизложенным, тема исследовательской работы направлена на изучение и разработку модели нейронной сети, распознающей образы, а конкретно – цифры.

Актуальность данной программы обусловлена широким применением подобных технологий и, вследствие этого, большой популярностью на рынке программ, распознающих образы.

Объект исследования – распознавание цифр

Предмет исследования – искусственные нейронные сети

Цель исследования – разработка программы по распознаванию цифр методом нейросетевого моделирования

Задачи исследования:

1. Изучить принцип работы нейронных сетей
2. Написать программу, распознающую образы цифр

Методы исследования: анализ, синтез, дедукция, эксперимент

# **Нейросети**

## **Краткая история ИНС**

Исследования в области искусственных нейронных сетей пережили несколько периодов активизации.

Первый период был в 1943 году, когда нейрофизиолог Уоррен Мак-Каллох и математик Уолтер Питтс написали статью о работе нейронов. Для описания работы нейронов мозга, они смоделировали простую НС, с использованием электрической цепи. Их основная идея заключалась в том, что любая связь типа "вход-выход" может быть реализована искусственной (формальной) НС.

Вторым этапом в развитии НС стало изобретение перцептрона в 1957 году Фрэнком Розенблаттом. В 1958 г. Фрэнк Розенблатт продемонстрировал компьютерную модель электронного устройства, названную им перцептроном, а в 1960 г. – первый действующий нейрокомпьютер «Марк-1», который моделировал совместную работу человеческого глаза и мозга. Основным назначением машины было распознавание. Концом этого этапа стала публикация Марвина Ли Минского и Сеймура Пайперта в 1969 году, в которой они указали на важный класс задач, которые однослойный перцептрон решать не может. Минский и Пайперт дискредитировали исследования НС и финансирование в области искусственного интеллекта. Но несмотря на демонстрацию Минским и Пайпертом ограничений перцептронов, исследования нейронной сети все же продолжались.

В 1982 году Джон Хопфилд – физик с мировым именем, заинтересовавшись нейронными сетями, написал две особо читаемые статьи о НС и провел многочисленные лекции по всему миру, чем убедил сотни высококвалифицированных ученых, математиков и технологов присоединиться к формирующемуся полю НС. Хопфилд показал, что высокосвязная сеть нейронов с обратными связями может быть описана как динамическая система, обладающая "энергией". При ассоциативном вызове сеть, стартующая в произвольном (случайном) состоянии, сходится к конечному устойчивому состоянию с наименьшей энергией. Новый подход к описанию НС с обратными связями оказался очень плодотворным.

Подобный прорыв произошел и в связи с многослойными сетями без обратных связей. Для обучения таких сетей был разработан алгоритм обратного распространения ошибки.

С середины 80-х годов теория нейронных сетей получила «технологический импульс», вызванный появлением новых доступных и высокопроизводительных персональных компьютеров.

Самых значительных достижений в данном вопросе достигла американская компания IBM (англ. Internationa Business Machines). Первые результаты исследований были продемонстрированы 14 ноября 2009 года. Компания представила на суд общественности успешно смоделированный мозг кошки. Правда, следует отметить, что тогда его работа была в 643 раза медленнее реального времени.

Следующей вехой в развитии ИНС можно считать 18 августа 2011 года, когда IBM создали передовой на тот момент нейронный процессор, который содержал 256 нейронов и 262144 синапсов.

## **1.2 Искусственный нейрон**

Искусственные нейронные сети являются относительно грубыми электронными моделями, основанными на нервной структуре головного мозга. Фундаментальным обрабатывающим элементом нейронной сети является нейрон. Нейрон (нервная клетка) — это специальная биологическая клетка, которая обрабатывает информацию. По данным оценки, в мозге существует огромное количество нейронов, каждый из которых обладает примерно 1011 многочисленными взаимосвязями.

Изображение выглядит как текст, карта

Описание создано с очень высокой степенью достоверности

Рисунок . Биологический нейрон

Как показано на приведенной выше схеме, типичный нейрон состоит из следующих четырех частей, с помощью которых мы можем объяснить его работу:

* Дендриты — древоподобные ветвей, отвечающий за получение информации от других нейронов, к которым подключен данный. В другом смысле, мы можем сказать, что это «уши» нейрона;
* Ядро — это тело клетки нейрона, отвечающее за обработку информации, полученной от дендритов;
* Аксон — некий «кабель», посредством которого нейроны посылают информацию;
* Синапсы — это соединения между аксоном и дендритами других нейронов.

Последние экспериментальные данные предоставили дополнительные доказательства того, что нейроны структурно более сложно устроены, чем упрощенно описано выше. Они значительно более сложны, чем существующие искусственные нейроны, которые встроены в сегодняшние искусственные нейронные сети. По мере того, как биология обеспечивает лучшее понимание нейронов, и по мере развития технологий, разработчики сетей могут продолжать совершенствовать свои системы, опираясь на понимание человеческого мозга.

Но в настоящее время грандиозное воссоздание мозга – не является целью искусственных нейронных сетей. Наоборот, нейросетевые исследователи ищут способы применения возможностей природы, для которых люди могут спроектировать решения проблем, которые не были решены с помощью традиционных вычислений.

Для этого нужна основная единица нейронных сетей – искусственные нейроны, моделирующие четыре основные функции биологических нейронов. Рисунок 2 показывает фундаментальное представление искусственного нейрона - перцептрона.



Рисунок 2. Искусственный нейрон

|  |  |
| --- | --- |
| Биологическая нейронная сеть | Искусственная нейронная сеть |
| Ядро | Узел |
| Дендриты | Входные данные |
| Синапсы | Веса |
| Аксон | Выходные данные |

Таблица 1 – Сравнительная характеристика понятий БНС и ИНС

### **Перцептрон**

Перцептрон является абстрактной моделью биологического нейрона. По сути, это очень простой процессор.

Перцептрон состоит из трёх типов элементов, а именно: поступающие от датчиков сигналы передаются ассоциативным элементам, а затем реагирующим элементам. Таким образом, перцептроны позволяют создать набор «ассоциаций» между входными стимулами и необходимой реакцией на выходе. В биологическом плане это соответствует преобразованию, например, зрительной информации в физиологический ответ от двигательных нейронов.

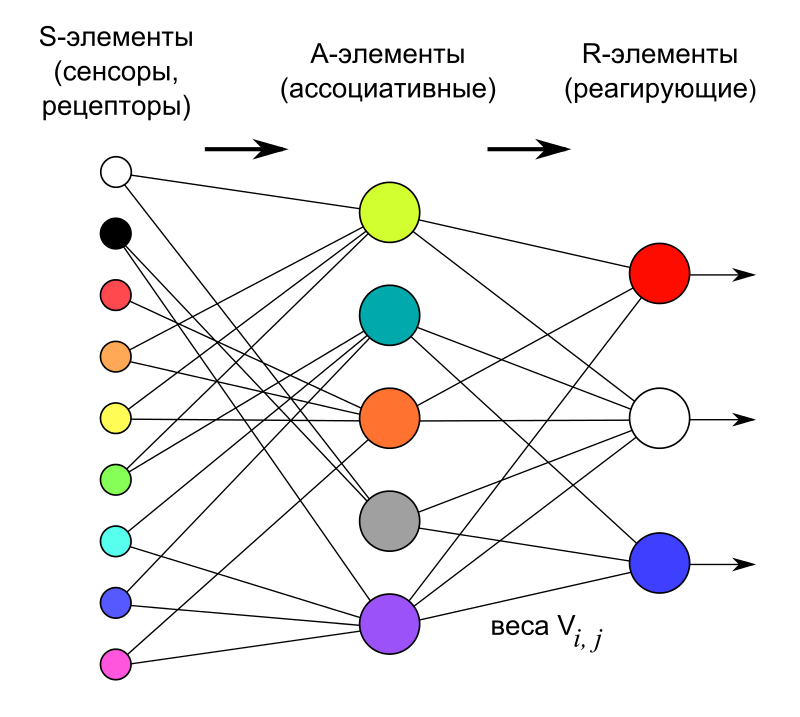


Рисунок 2. Логическая схема перцептрона с тремя выходами

Как же он работает? Как можно увидеть по рисунку, перцептрон принимает несколько двоичных входов, x1, x2,...,xn и производит один двоичный выход. Ранее упомянутый Розенблатт предложил простое правило для вычисления выходных данных. Он ввел веса, w1, w2,...,wn – вещественные числа, выражающие важность соответствующих входных данных для вывода. Выход нейрона от того, является ли результат работы сумматора – сумма , называемая взвешенной, меньше или больше некоторого порогового значения. Так же, как и веса, пороговое значение является вещественным числом, являющимся параметром нейрона. Опишем это в более точных алгебраических терминах, как в формуле (1):

(1)

Это основная математическая модель. Вы можете думать о перцептроне так: это устройство, которое принимает решения, взвешивая различные виды данных. Поэтому должно быть очевидным, что сложная сеть перцептронов может принимать довольно сложные решения.

Рассмотрим сеть перцептронов, изображенную на рисунке 3. В этой сети первая колонка перцептронов - то, что мы будем называть входным слоем перцептронов – принимает простые решения, взвешивая входные данные. Каждый из перцептронов второго слоя принимает решение, взвешивая выходные данные перцептронов входного слоя. То есть, перцептрон во втором слое может принимать решение на более сложном и более абстрактном уровне, чем перцептроны в первом слое. И еще более сложные решения может принимать перцептрон в третьем слое и т.д. Таким образом, многослойная сеть перцептронов может участвовать в принятии сложных решений.

Обучающие алгоритмы автоматически настраивают веса и отклонения сети нейронов в ответ на внешний стимул, без прямого участия программиста. Процесс обучения предполагает малое изменение весов и отклонений, чтобы получить малое изменение на выходе, приближаясь к лучшему результату классификации. Но сеть персептронов так не работает, и даже малое воздействие может вызвать резкие изменения, приводя к непредсказуемым последствиям. Для решения этой проблемы используют так называемые функции активации. Для перцептрона функцией активации является единичная функция.

### **1.2.2 Функция активации**

### Функция, которая получает входной сигнал, но генерирует выходной сигнал с учетом порогового значения, называется функцией активации. С математической точки зрения существует множество таких функций, которые могли бы обеспечить подобный эффект. В качестве примера можно привести ступенчатую функцию.

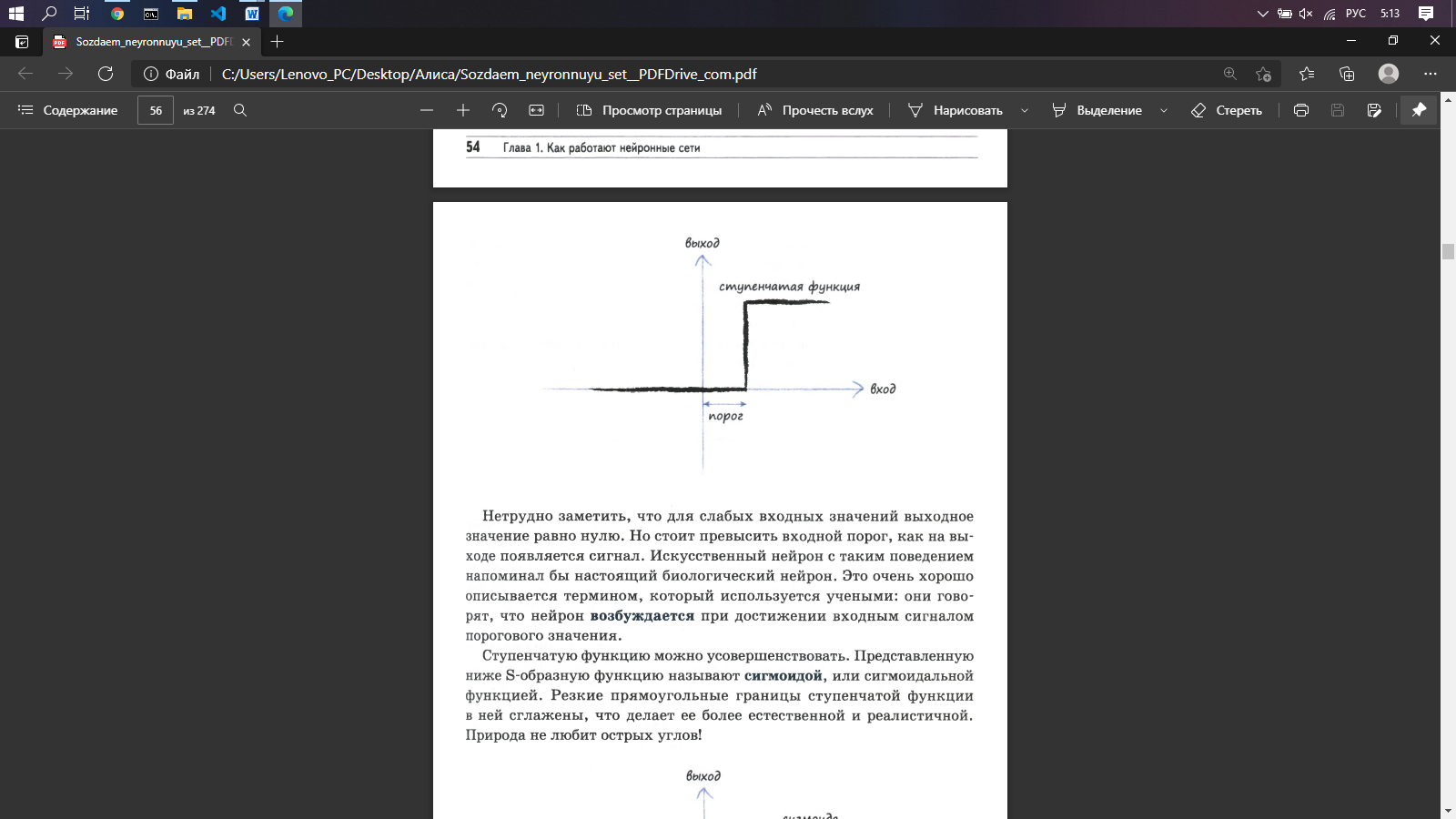


Рисунок 3. Ступенчатая функция

Нетрудно заметить, что для слабых входных значений выходное значение равно нулю. Но стоит превысить входной порог, как на выходе появляется сигнал. Искусственный нейрон с таким поведением напоминал бы настоящий биологический нейрон. Это очень хорошо описывается термином, который используется учеными: они говорят, что нейрон возбуждается при достижении входным сигналом порогового значения. Ступенчатую функцию можно усовершенствовать. Представленную ниже S-образную функцию называют сигмоидой, или сигмоидальной функцией. Резкие прямоугольные границы ступенчатой функции в ней сглажены, что делает ее более естественной и реалистичной.

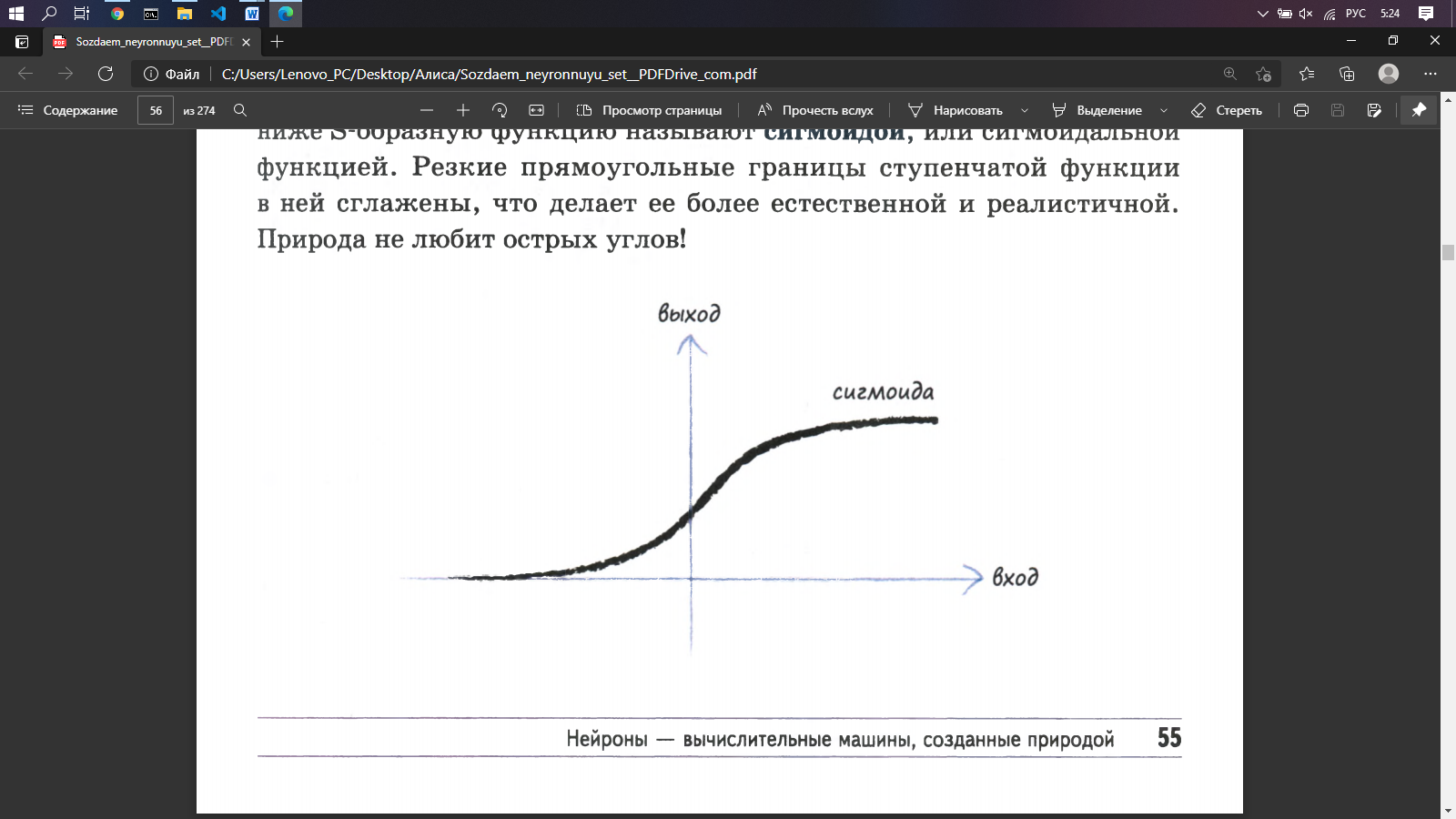


Рисунок 4. Сигмоида

Сигмоида, которую иногда называют также логистической функцией, описывается следующей формулой (2):

## **1.3 Архитектура нейронных сетей**

Существует множество видов нейронных сетей.

Сеть прямого распространения сигнала (сеть прямой передачи) - нейронная сеть без обратных связей. В этой сети распространение сигнала однонаправлено, то есть нет обратных связей. От входного слоя сигнал обрабатывается слой за слоем в направлении выхода. Через известное число шагов на выходном слое появляется ответ сети.

Сети прямого распространения являются хорошо изученными и относительно простыми в реализации. Их недостатком является необходимость большого числа нейронов для выполнения сложных задач.

Цепи Маркова — своего рода предшественники машин Больцмана и сетей Хопфилда, о которых сказано ниже. В цепях Маркова мы задаем вероятности перехода из текущего состояния в соседние. Кроме того, это цепи не имеют памяти: последующее состояние зависит только от текущего и не зависит от всех прошлых состояний. Хотя цепь Маркова нельзя назвать нейронной сетью, она близка к ним и формирует теоретическую основу для машин Больцмана и сетей Хопфилда.

Сеть Хопфилда (модель Хопфилда), является одним из видов сетей ассоциативной памяти. Это однослойная нейронная сеть, в которой каждый нейрон связан со всеми остальными, имеет по одному входу и выходу. Жесткая функция активации генерирует два значения: -1 (заторможен) и +1 (возбужден). В модели используется принцип хранения информации как динамически устойчивых аттракторов. Энергетическая функция уменьшается в процессе обучения пока не достигает локального минимума (аттрактора), в котором сохраняет постоянное значение.

Машины Больцмана во многом похожи на сети Хопфилда, но в них некоторые нейроны помечены как входные, а некоторые остаются скрытыми. Входные нейроны становятся выходными, когда все нейроны в сети обновляют свои состояния. Сначала весовые коэффициенты присваиваются случайным образом, затем происходит обучение методом обратного распространения, или в последнее время все чаще с помощью алгоритма contrastive divergence (когда градиент вычисляется при помощи марковской цепи). Машины Больцмана — стохастическая нейронная сеть, так как в обучении задействована цепь Маркова.

Рекуррентные сети — это глубокие сети, в которых присутствуют обратные связи. Это значит, что присутствует хотя бы один слой, сигналы с которого поступают на него же, либо на один из предыдущих слоев. Нейроны участвуют в обработке информации многократно, что позволяет использовать динамические свойства сети. Такие сети позволяют сократить число нейронов. На основе рекуррентных сетей разработаны различные модели ассоциативной памяти. Особенно эти сети пригодились в области распознавания речи.

Сеть Хемминга (Классификатор по минимуму расстояния Хемминга) - другое пример нейронной сети ассоциативной памяти. Принцип работы основан на вычислении расстояния Хемминга от входного вектора до всех векторов-образцов, известных сети. При поступлении входного образа, сеть выбирает образец с наименьшим до него расстоянием Хемминга и соответствующий ему выход активизируется.

Сеть Кохонена - однослойная сеть из настраиваемых весов. В то время как один нейрон возбуждается, все остальные выходы слоя подавляются. Веса подстраиваются так, что входные образы из одного класса активируют один и тот же выходной нейрон. Так входные вектора классифицируются по схожим группам. Это отражает одно из важнейших свойств сети Кохонена - способность к обобщению. Вектор каждого из нейронов сети заменяет группу соответствующих ему классифицируемых векторов.

Глубокие сети доверия — это сети, представляющие собой каскад Ограниченных Машин Больцманна. Стандартная Машина Больцманна состоит из полносвязных "видимых" и "скрытых" нейронов, принимающих бинарные значения, определенные векторами. Ограниченные машины отличаются тем, что нейроны одного класса не связаны между собой. Эти сети интересны тем, что могут играть роль генерирующих моделей. Иными словами, сеть обученная распознавать, например, рукописный текст в теории может быть использована для генерации изображений, которые выглядят как рукописный текст.

Свёрточные нейронные сети и глубокие свёрточные нейронные сети кардинально отличаются от других сетей. Они используются в основном для обработки изображений, иногда для аудио и других видов входных данных. Типичным способом их применения является классификация изображений. Такие сети обычно используют «сканер», не обрабатывающий все данные за один раз.

## **1. 4 Методы обучения ИНС**

Каждая нейронная сеть требует обучения, в противном случае правильный результат вряд ли будет получен. **Обучение нейронной сети** (Training)— это поиск такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в нужный нам выходной.

Рассмотрим некоторые из методов обучения нейронных сетей.

1. Метод обратного распространения ошибки.

Этот метод является основным и имеет ещё одно название — Backpropagation, так как использует алгоритм градиентного спуска. То есть при помощи движения вдоль градиента рассчитывается локальный минимум и максимум функции.

В процессе поступления информации нейронная сеть последовательно передаёт её от одного нейрона к другому посредством синапсов, до того момента, пока информация не окажется на выходном слое и не будет выдана как результат. Такой способ называется передачей вперёд. После того, как результат получен вычисляется ошибка и на её сновании выполняем обратную передачу. Суть, которой — последовательно изменить вес синапсов начиная с выходного и продвигаясь к входному слою. При этом значение веса меняется в сторону лучшего результата. Для использования такого метода обучения подойдут только те функции активации, которые можно дифференцировать. Так как обратное распространение вычисляется с помощью высчитывания разницы результатов и умножения его на производную функции от входного значения. Для того, чтобы успешно провести обучение, необходимо распространить полученную ошибку на весь вес сети. Высчитав ошибку на выходном уровне, а также там можно вычислить дельту, которая будет последовательно передаваться между нейронами. Затем необходимо произвести расчёт градиента для каждой исходящей связи. Затем имея все необходимые данные необходимо выполнить обновление весов и рассчитать благодаря функции метода значение, которое станет величиной изменения. При этом не стоит забывать про момент и скорость обучения.

Одна итерация этого метода даёт небольшой процент уменьшения ошибки, поэтому повторять их необходимо снова и снова пока показатель ошибки не будет приближен к 0.

1. Метод упругого распространения

Предыдущий способ обучения, представленный выше, имеет недостаток в виде больших временных затрат на процесс обучения неуместных в случае необходимости получить быстрый результат. Для ускорения процесса было предложено немало дополнительных алгоритмов, ускоряющих процесс. Одним из которых и является текущий метод. Настоящий алгоритм использует в качестве основы обучение по эпохам и применяет только знаки производных частного случая для корректировки весовых коэффициентов. Используется определённое правило, по которому производится расчёт величины коррекции весового коэффициента:

Если на этом этапе расчётов производная меняет свой знак, значит, изменение было слишком большим и локальный минимум был упущен и нужно произвести откат, то есть вес вернуть в обратную позицию, а величину изменения уменьшить.

Если знак производной не изменился, то величина изменения веса, наоборот, увеличивается для большей сходимости.

Если основные параметры коррекции веса зафиксировать, то настройки глобальных параметров можно избежать. И это станет ещё одним преимуществом текущего метода над предыдущим. Для этих параметров есть рекомендуемые значения, однако, никаких ограничений на их выбор не накладывается.

Чтобы вес не принимал слишком большие или малые значения используются установленные ограничения величины коррекции. Значение коррекции также вычисляется по определённому правилу:

То есть если производная функции в конкретной точке меняет знак с плюса на минус, означает, что ошибка возрастает и вес требует коррекции и происходит его уменьшение, в противном случае — увеличение.

Такой подход позволяет добиться сходимости нейросети быстрее в несколько раз в отличие от предыдущего варианта обучения.

1. Генетический алгоритм

Третий наиболее интересный алгоритм обучения искусственных нейронных сетей — Genetic Algorithm. Он представляет собой упрощённую интерпретацию природного алгоритма, основанного на скрещивании результатов. То есть, по сути, происходит скрещивание результатов, выбор наилучших и формирование на их основе нового поколения.

В случае если результат не устраивает алгоритм повторяется пока поколение не становиться идеальным. Алгоритм может завершиться без достижения нужного результата если количество попыток будет исчерпано или же будет исчерпан время на мутацию. Этот алгоритм применим к процессу оптимизации веса нейронной сети, при заданной по умолчанию топологии.

При этом вес кодируется двоичным кодом и каждый результат определяется полным набором веса. Оценка качества происходит методом вычисления ошибки на выходе.

# **Распознавание образов**

## **Принципы распознавания образов**

Основным свойством живых организмов считается способность «распознавать», т.е. выявлять в потоке информации, которая поступает от органов чувств, определенные объекты, закономерности и явления. Образ представляет собой описание объекта. Во время нашего бодрствования мы ежесекундно совершаем акты распознавания. Мы распознаем окружающие нас образы и, в соответствии с этими образами, мы перемещаемся и совершаем определенные действия. Без особого труда мы можем заметить в толпе знакомого нам человека и разобрать, что он говорит, можем узнать голос друга, прочитать напечатанный или рукописный текст и идентифицировать отпечатки пальцев, можем отличить злобную гримасу от улыбки. Человек представляет собой неимоверно сложную информационную систему – в некоторой степени это определяется крайне развитыми способностями человека распознавать образы.

Распознавание человеком конкретных образов можно рассматривать как психофизиологическую задачу, связанную с процессом взаимодействия индивида с определенным физическим раздражителям. Фактически, распознавание человеком образов основывается на схожести однотипных объектов. То есть распознавание сводится к оценке относительных шансов на то, что исходные данные соответствуют тому или иному из известных множеств статистических совокупностей, определяющихся прошлым опытом человека и предоставляющих ориентиры и априорную информацию для распознавания. Таким образом, задачу распознавания образов можно приравнять к задаче определения различий между исходными данными причем не с помощью отождествления с отдельными образами, а с их совокупностями; последнее осуществляется при помощи поиска признаков (инвариантных свойств) на множестве объектов, образующих определенную совокупность.

Иначе говоря, распознавание образов можно определить как отнесение исходных данных к определенному классу образов с помощью выделения существенных признаков или свойств, характеризующих эти данные, из общей массы несущественных деталей. Под классом образов понимается некоторая категория, определяющаяся рядом свойств, общих для всех элементов. Под образом понимают описание любого элемента как представителя соответствующего класса образов.

Очевидное решение задачи распознавания образов заключается в применении к отдельным предъявленным образам ряда простых тестов для выявления признаков каждого класса. Совокупность всех этих тестов должна распознавать все допустимые образы из разных классов.

Если следовать такому интуитивному подходу, то построение автоматической системы распознавания образов может показаться достаточно простой задачей. Однако, не существует общей теории, которая бы позволяла определить, какие из всего множества возможных тестов следует применить к предъявленным образам. И подобный подход чрезмерно зависит от опыта и технической интуиции разработчика и поэтому часто не дает удовлетворительного решения задач распознавания образов, встречающихся в практической деятельности.

Но вернемся к конкретной задаче - распознаванию рукописных цифр. Каждая конкретная цифра вне зависимости от того, как она написана, обладает рядом общих признаков, которые и будут использоваться в качестве средств её идентификации. Цифры идентифицируются и классифицируются согласно отмеченным у них признакам. Значит, основные функции системы распознавания образов заключаются в обнаружении и выделении общих признаков у рукописных цифр, которые принадлежат к одному и тому же классу образов, т.е. являются одной и той же цифрой, узнавании этой цифры в любой другой обстановке и отнесении её к одному из заданных десяти классов: цифра «0», цифра «1», цифра «2», цифра «3», цифра «4», цифра «5», цифра «6», цифра «7», цифра «8», цифра «9».

## **Система распознавания образов**

Системы распознавания имеют следующую типичную функциональную схему, представленную на рисунке 6.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

Рисунок 5.Функциональная схема распознающих систем

*Входные данные*, которые подлежат распознаванию, подаются на вход системы и подвергаются предобработке для их дальнейшего преобразования в необходимый для следующего этапа вид или для выделения из них необходимых отличительных черт. В последующем на этапе *принятия решения* над обработанным массивом данных производится ряд вычислений и на основе их результатов формируется ответ, содержащий ожидаемые от системы сведения о входных данных. Содержание входных и *выходных данных* определяется назначением системы. Например, если входом служит изображение рукописной цифры, то в качестве выходных данных система может выдать название цифры.

Кроме описанных этапов функционирования системы распознавания предусматривают свою настройку на множество возможных входных данных; этот этап называют этапом *обучения* системы. Целью обучения системы является формирование в её памяти набора сведений, необходимых для распознавания предполагаемого класса входных данных.

На этапе *предобработки* решается задача формирования формализованного описания объектов распознавания, подходящего для использования алгоритмами распознавания. Как правило, исходные данные о наблюдаемых объектах, поданных в форме, неподходящей для распознавания. Это могут быть растровые изображения, аудиофайлы, статистические данные (наборы чисел), видеозаписи и прочее. Некоторые алгоритмы распознавания требуют более высокоуровневого представления. Это приводит к необходимости произвести одно или более преобразований исходных данных.

Этап принятия решения является наиболее существенным в процессе работы системы распознавания с точки зрения её характеристики в целом. Т.е. задача, которая решается на данном этапе, во многом определяет назначение системы.

Рассмотрим два основных класса задач, решаемых на этапе принятия решений.

а)Распознавание — распределение предъявляемых объектов по определённым классам посредством применения известных правил классификации. Это одна из наиболее характерных задач для систем распознавания. Перед тем, как система сможет выполнять данную функцию, предполагается её обучение на множестве разнообразных примеров — обучающей выборке объектов распознавания.

б)Классификация — это разбиение множества объектов на непересекающиеся классы по их формализованным описаниям. Данная задача решается в случаях, когда от системы не требуется отнести входные образы к каким-либо определённым классам, а требуется лишь способность различать их каким-либо способом по определённым признакам.

Для решения задачи о распознавании рукописных цифр, имеет смысл рассматривать задачу классификации. Ведь при решении этой задачи от системы не требуется разбивать входные образы на определенные классы, а требуется лишь различать их по определенным признакам и определять какому из заданных классов они принадлежат.

## **Задача классификации образов**

Задачу классификации объекта нередко ставят следующим образом. Имеется некоторый набор входных объектов (в нашем случае – рукописных цифр), принадлежащих заранее известному конечному множеству классов Ck, где k =1,..,n, и некоторое конечное множество объектов (называемое обучающим множеством), про каждый из которых известно, какому классу он принадлежит. Нужно построить алгоритм, который по любому входному объекту, не обязательно принадлежащему обучающему множеству, решает, какому классу этот объект принадлежит. Качество распознавания оценивается как вероятность (т.е. частота) ошибки классификации на другом конечном множестве объектов с заранее известными ответами (тестовом множестве).

Эту же задачу можно сформулировать в терминах линейной алгебры. Если входные переменные обозначить за вектор x с компонентами xi, где i = 1,..,d, а выходные переменные за вектор yk, где k =1,..,n, то задача сводится к определению набора функций, которые отображают xi на yk, где теперь yk указывают на то, к какому из классов принадлежит xi. К примеру, входные данные xi могут быть назначены классу, выходное значение которого yk является наибольшим.

Но как было сказано ранее, невозможно определить подходящую форму для требуемого отображения, кроме как с помощью набора тестов. Поэтому отображение моделируется в виде некоторой математической функции, которая содержит ряд настраиваемых параметров, значения которых определяются с помощью полученных данных. Мы можем записать такую функцию в виде формулы (3):

yk = yk (**x**, **w**), (3)

где w = (w1,..,wW) – вектор параметров, называемых весами.

Важность нейронных сетей в данном контексте заключается в том, что они предлагают очень мощную и общую структуру для представления нелинейного отображения из нескольких входных переменных для нескольких выходных переменных. Процесс определения значений этих параметров на основе наборов данных называется обучением, и по этой причине этот набор данных обычно называют тренировочным набором.

# **Создание программы**

## **Распознавание цифр методом нейросетевого моделирования**

Представим, что перед вами лежит изображение с цифрой. Посмотрев на него, вы легко сможете понять, что это за цифра, даже если она будет нарисована не совсем разборчиво. Поставим такую же задачу перед компьютером и напишем программу для решения данной задачи.

Программа принимает на вход изображение любой цифры в виде массива 28\*28 пикселей, а на выходе выдаёт значение изображенной цифры (от 0 до 9).

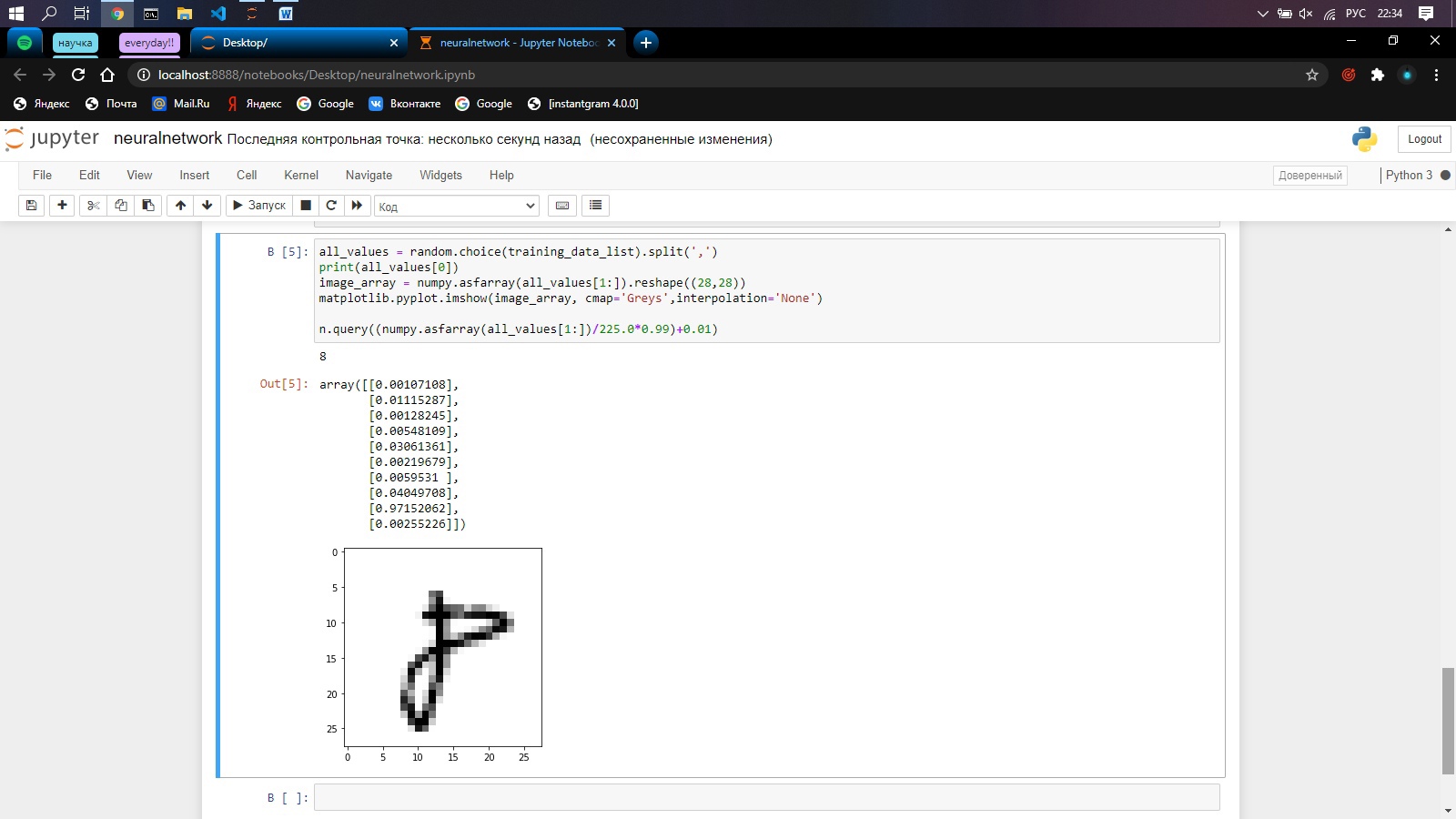
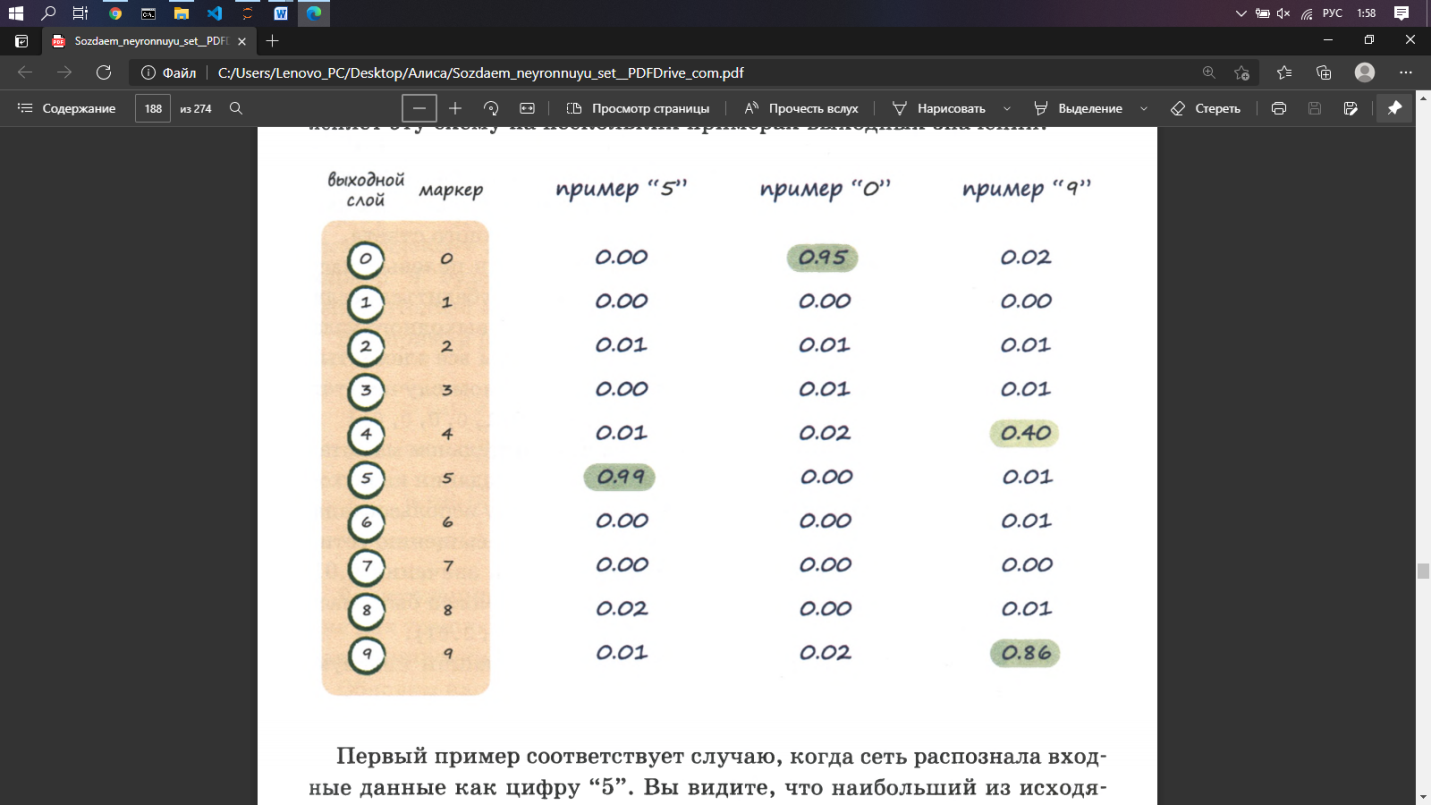
Например:

Рисунок 6. Пример изображения, подающегося на вход нейронной сети

Посчитав, мы получаем, что наша сетка состоит из 28\*28=784 пикселей, пусть есть 784 нейрона, содержащие различные числа от 0 до 1: чем ближе пиксель к белому цвету, тем ближе соответствующее число к единице. Эти заполняющие сетку числа назовем активациями нейронов.

Описанные 784 нейрона образуют первый слой нейросети - входной. Последний слой содержит 10 нейронов, каждый из которых соответствует одной из десяти цифр. В этих числах активация - число от нуля до единицы, отражающее насколько система уверена, что входное изображение содержит соответствующую цифру. (рис. 7)

Рисунок 7



Также есть средний слой, называемый скрытым, к рассмотрению которого мы вскоре перейдем. Выбор количества скрытых слоев и содержащихся в них нейронов произвольно, обычно они выбираются из определенных представлений о задаче, решаемой нейронной сетью, в данном случае будет использована трёхслойная НС, а количество нейронов в скрытом слое будет варьироваться по мере написания программы.

Для решения поставленной задачи необходимо будет выбрать алгоритм обучения ИНС, который будет находить значения весов и смещений.

Также необходимы будут тренировочные данные — последовательность данных, которыми оперирует нейронная сеть. Так как рассматриваемая задача очень распространена, для ее решения уже создана крупная база данных MNIST, состоящая из 60 тыс. размеченных данных и 10 тыс. тестовых изображений. Ею мы и воспользуемся.

## **Ход работы и отладка**

Для написания ИНС будем использовать Python 3.6, так как он лучше всего приспособлен для создания нейронных сетей. Помимо уже озвученной библиотеки MNIST, подключим также библиотеку NumPy для того, чтобы было легко и быстро производить необходимые алгебраические, и SkiPy - создана для работы с массивами NumPy и предоставляет множество удобных и эффективных численных методов, таких как процедуры численной интеграции и оптимизации.

Код класса нейронной сети можно разделить на три функции:

* инициализация — задание количества входных, скрытых и выходных узлов;
* тренировка — уточнение весовых коэффициентов в процессе обработки предоставленных для обучения сети тренировочных примеров;
* опрос — получение значений сигналов с выходных узлов после предоставления значений входящих сигналов.

После написания класса вводятся необходимые параметры (кол-во входных, скрытых, выходных узлов и коэффициент обучения НС ), а после загружаются тренировочные данные MNIST. Запускается тренировка сети на данных.

После каждого прохождения обучения будем изменять веса так, чтобы уменьшить размер ошибки, если результат расходится с ожидаемым. Таким образом, процесс обучения НС будет сводиться к уменьшению ошибки.

После тренировки НС на тренировочных данных, запускается её тестирование уже на тестовом наборе данных MNIST.

В зависимости от количества правильных ответов НС в сравнении с изначально известными верными значениями, выводится эффективность нейронной сети в процентах. На данный момент максимальная достигнутая мной эффективность НС достигает **97, 89%**.  
Также, я вывожу изначально известный правильный ответ, изображение 28\*28 пикселей и ответы НС в виде массива с показателем схожести цифры на изображении с каждым из маркером (от 0 до 9):

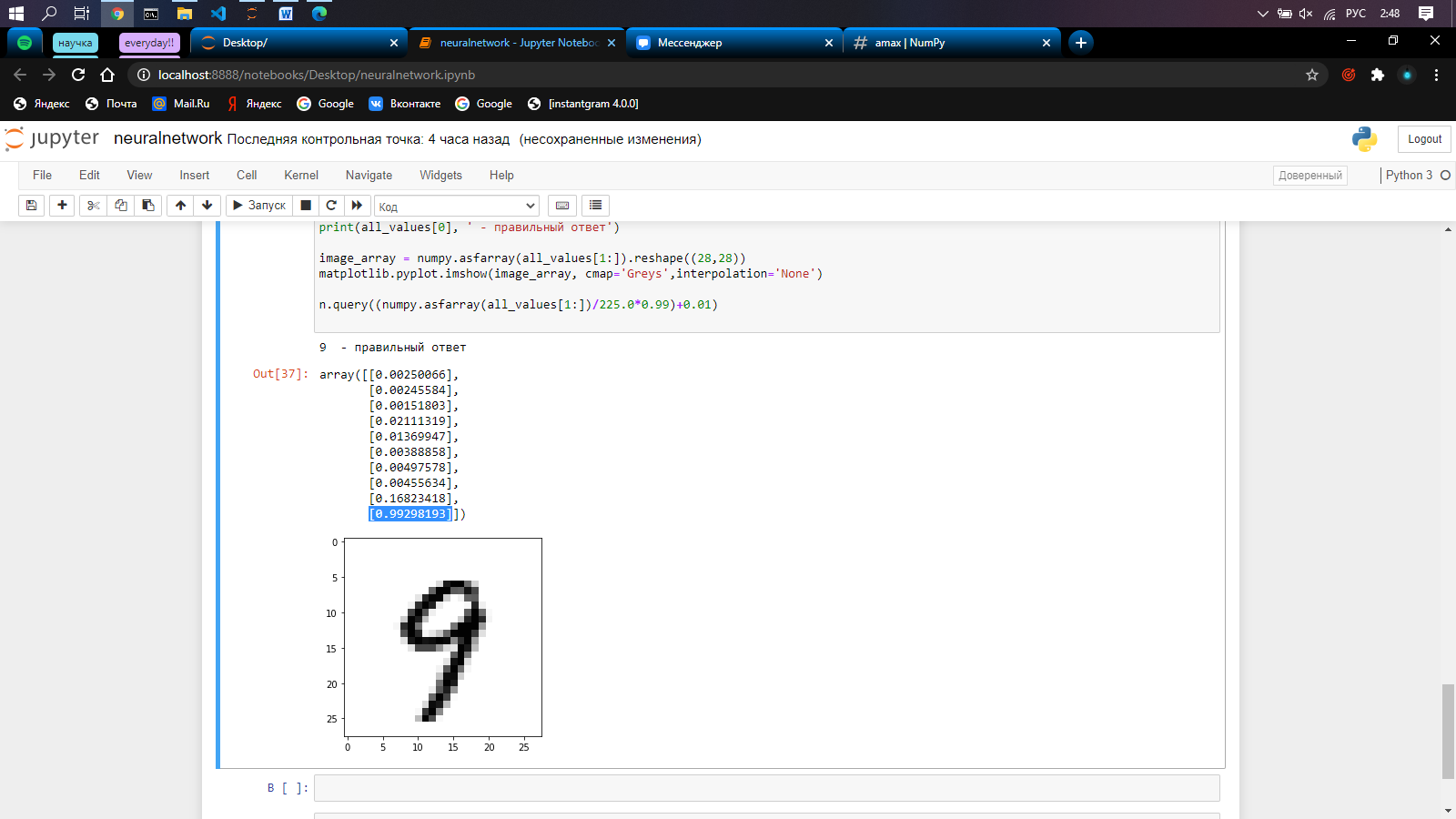
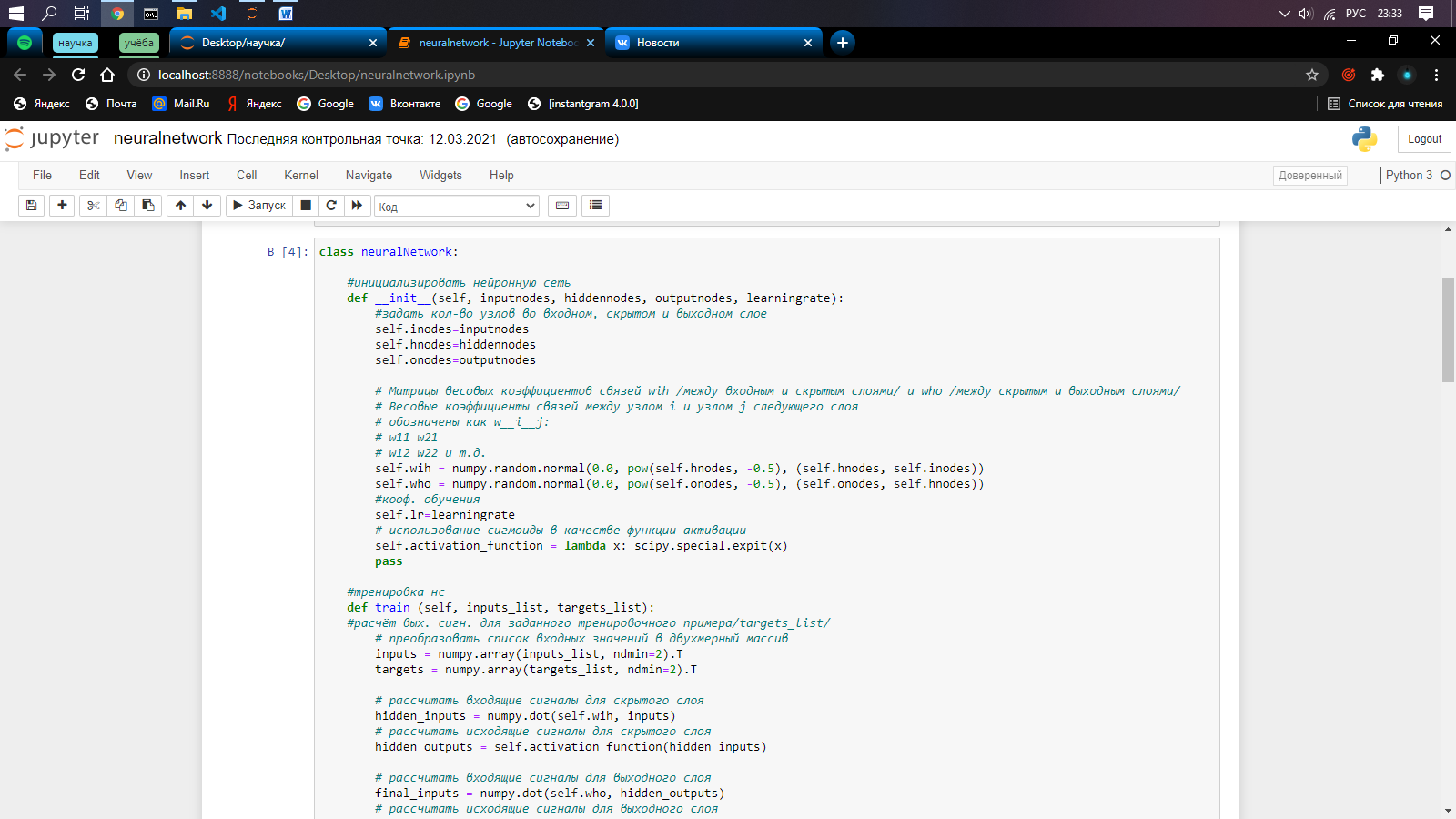
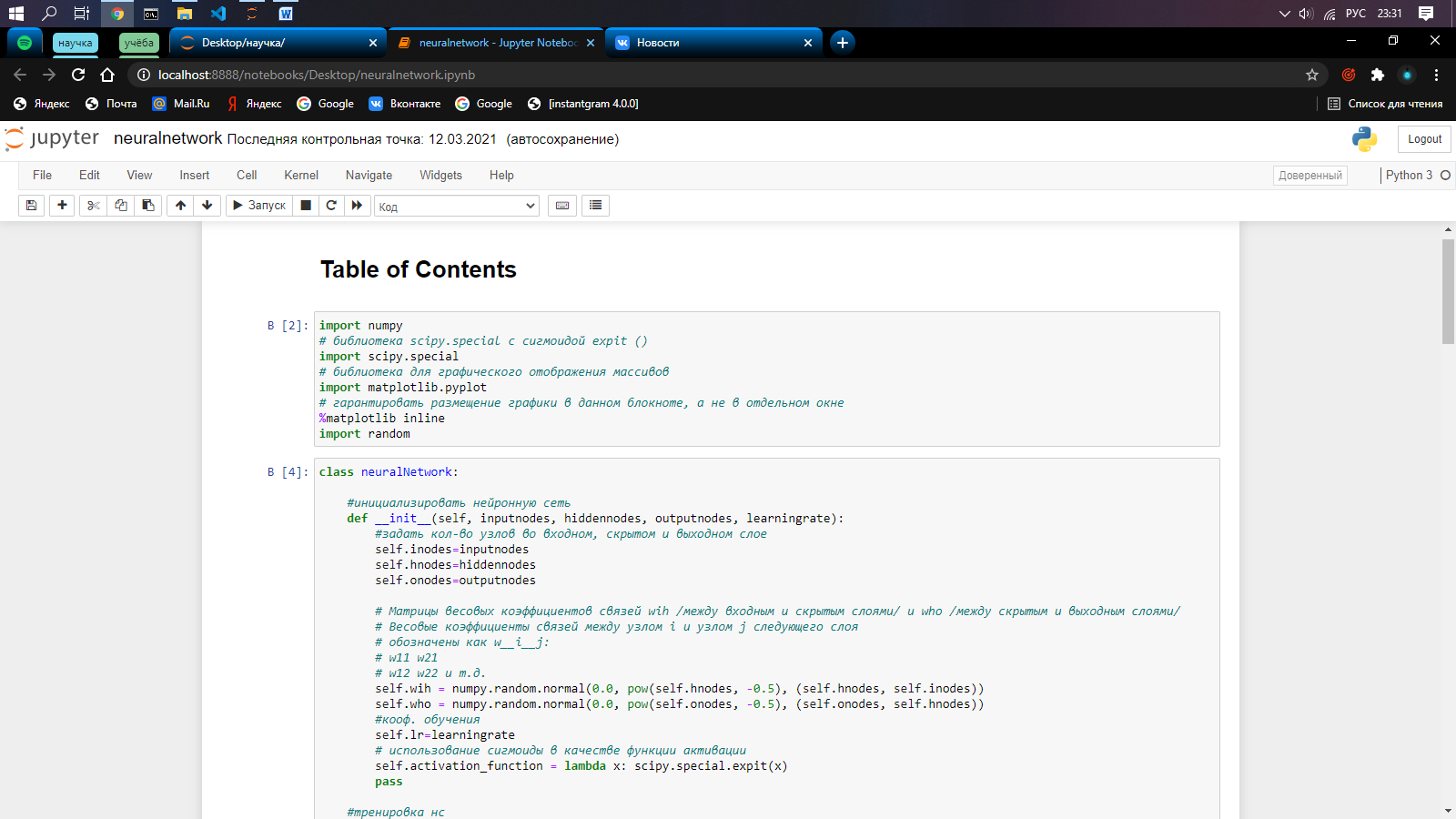
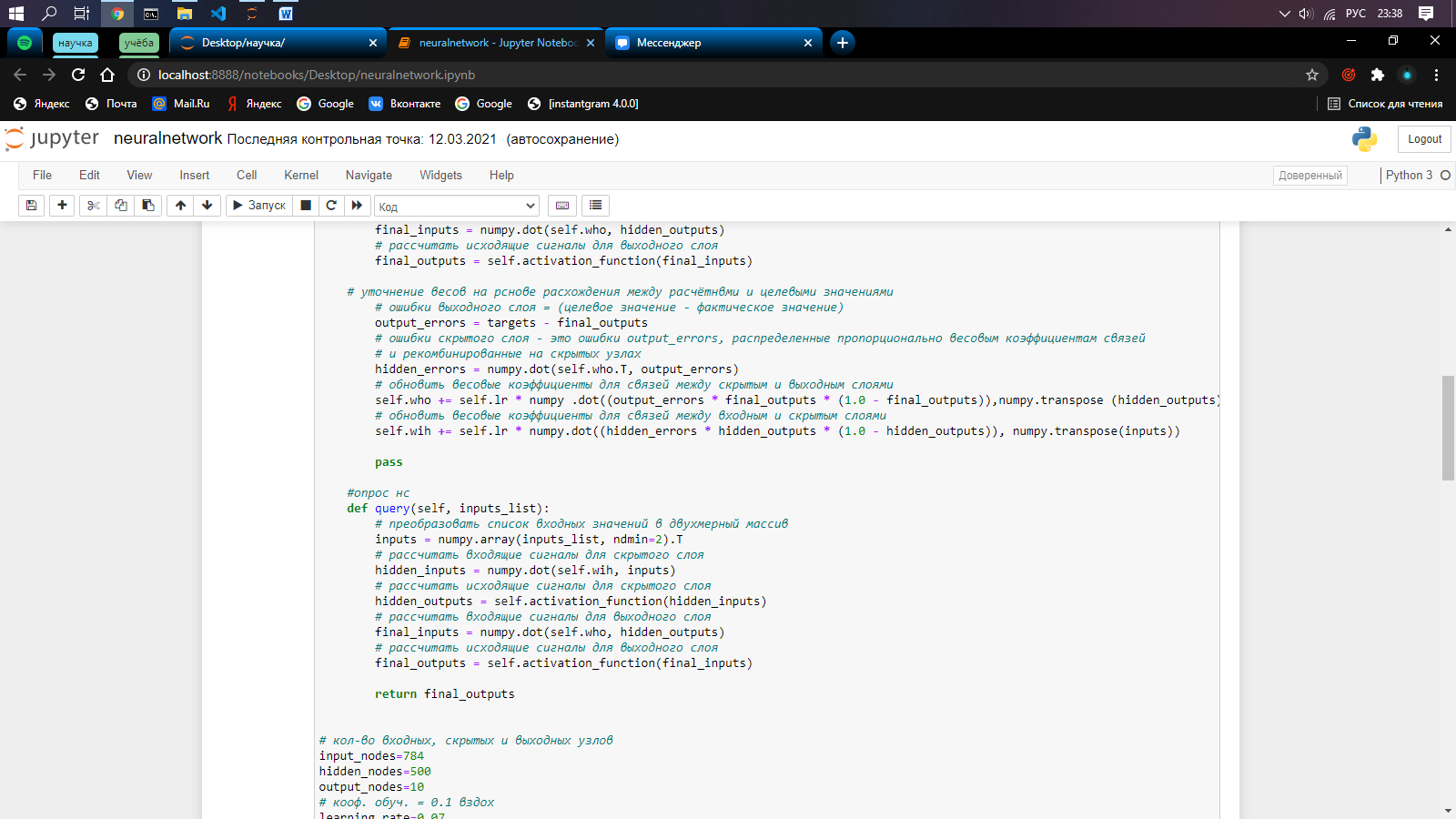
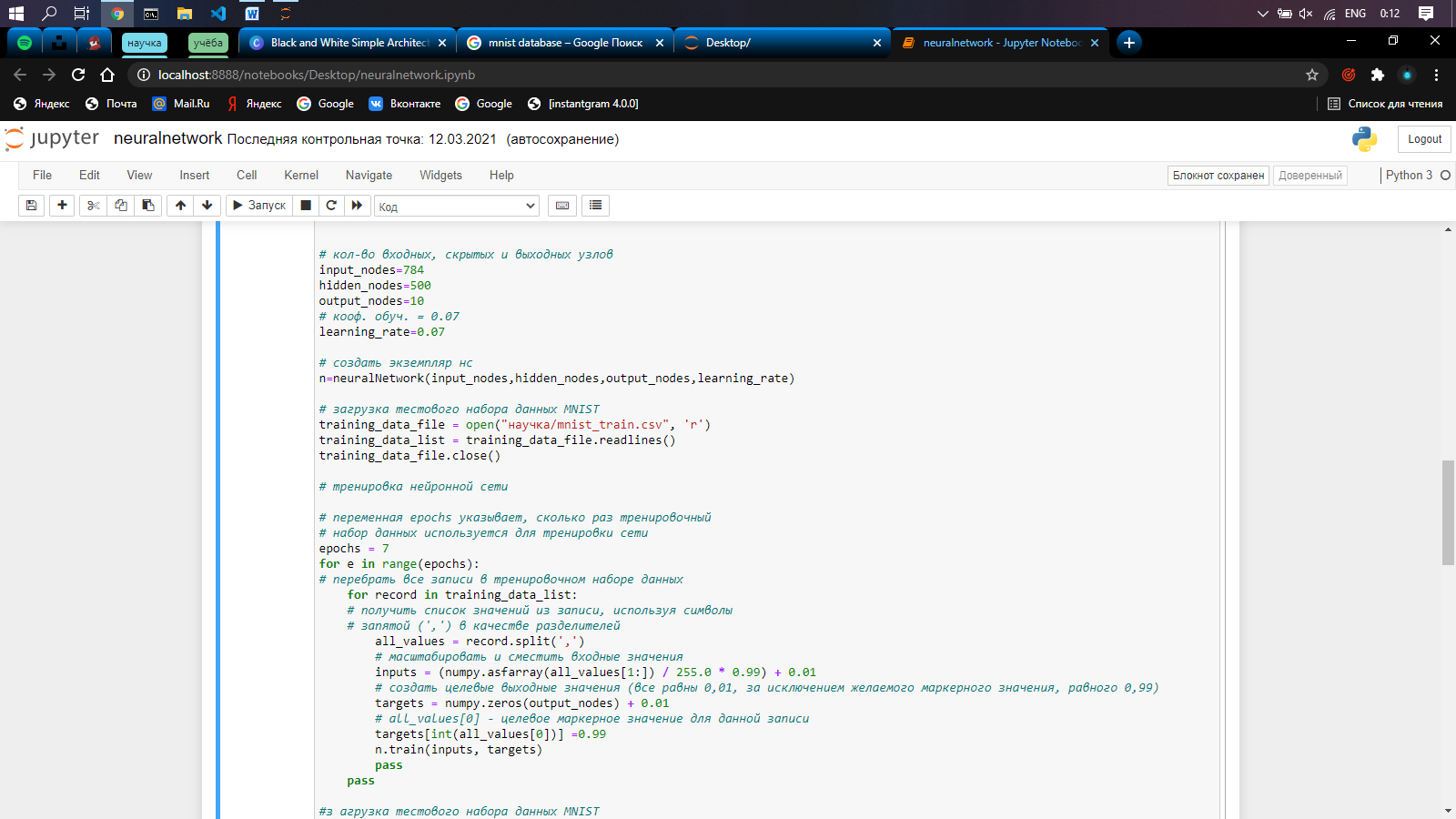


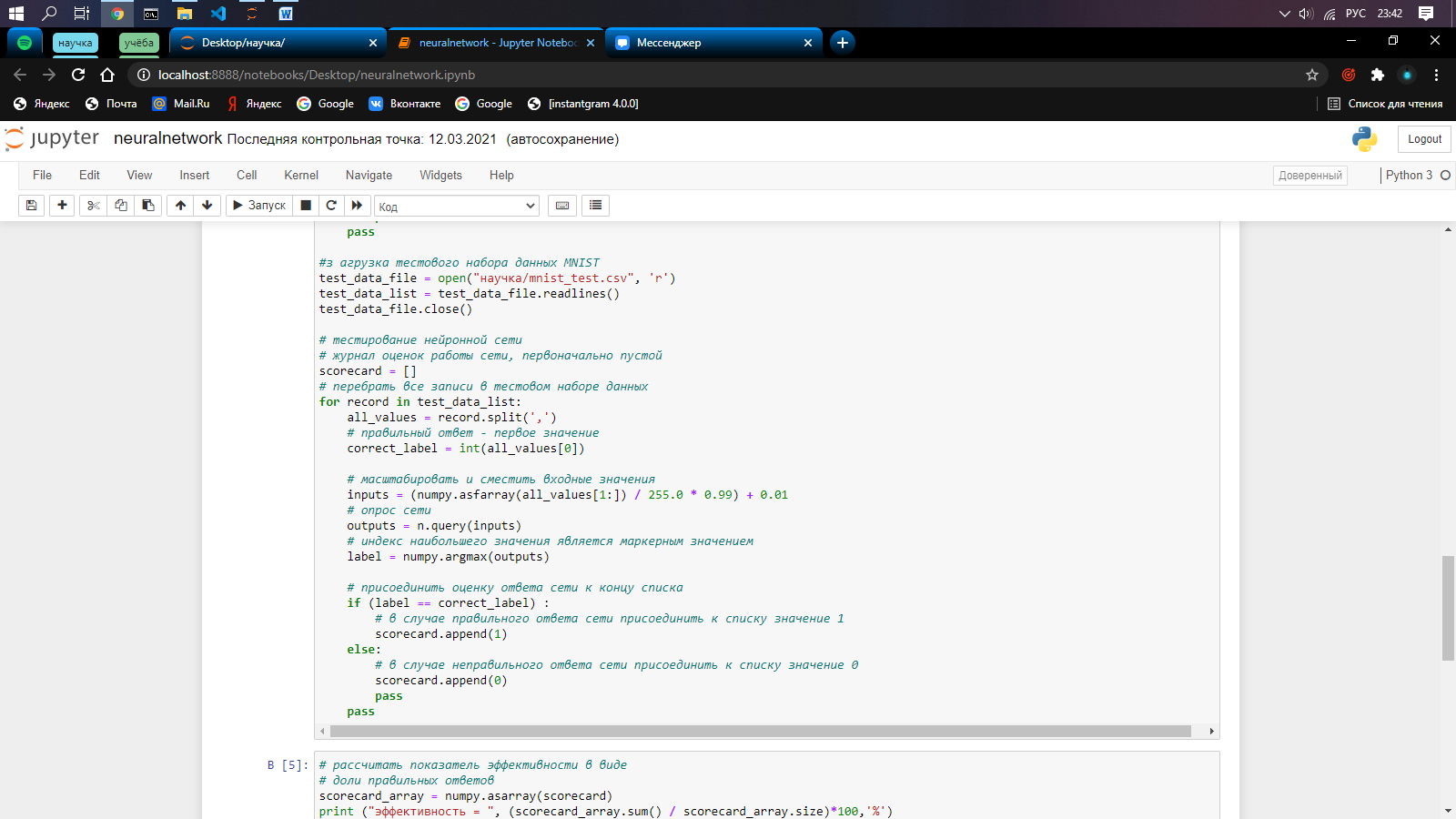
Рисунок 8. Пример выходных данных.

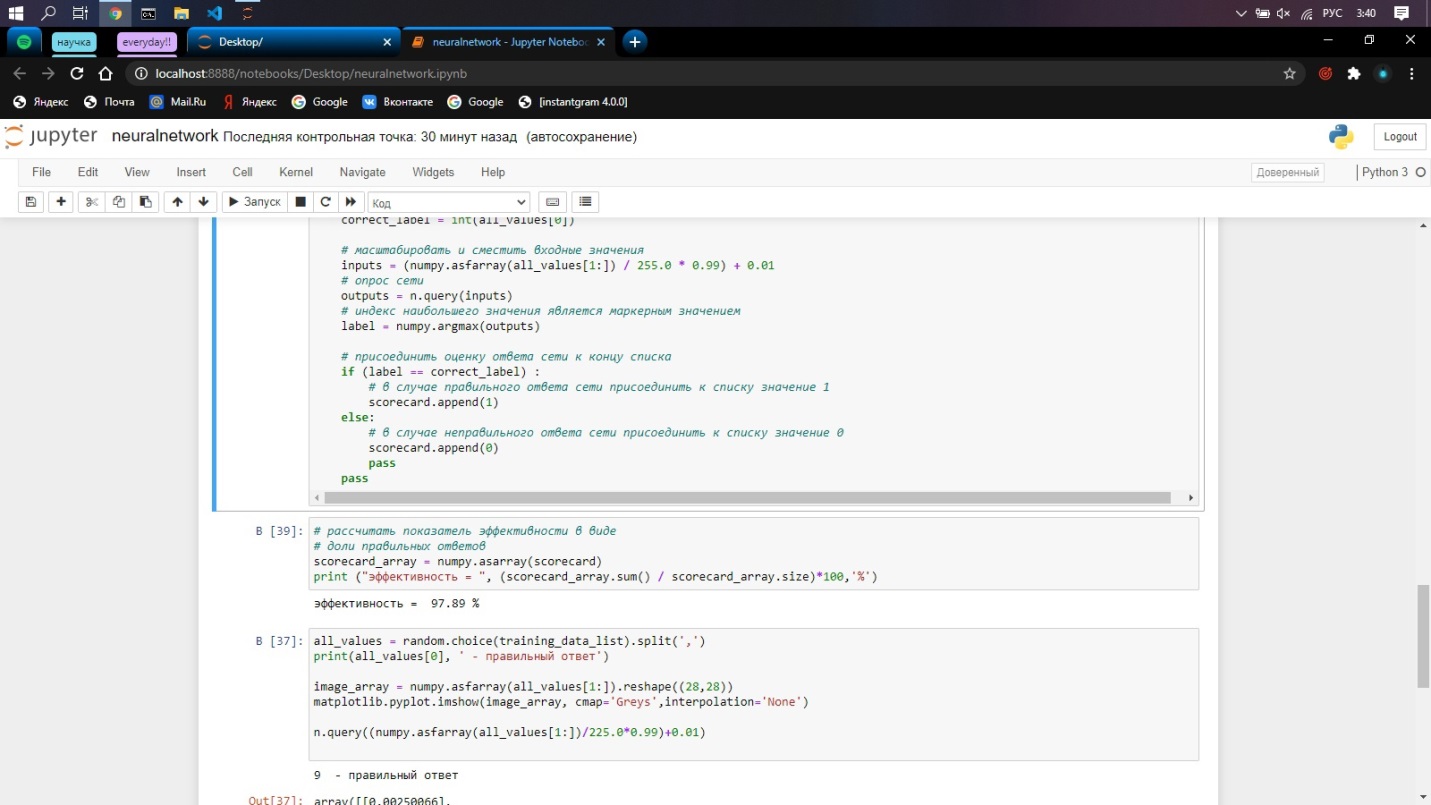
## **Код программы и примеры результата выполнения**

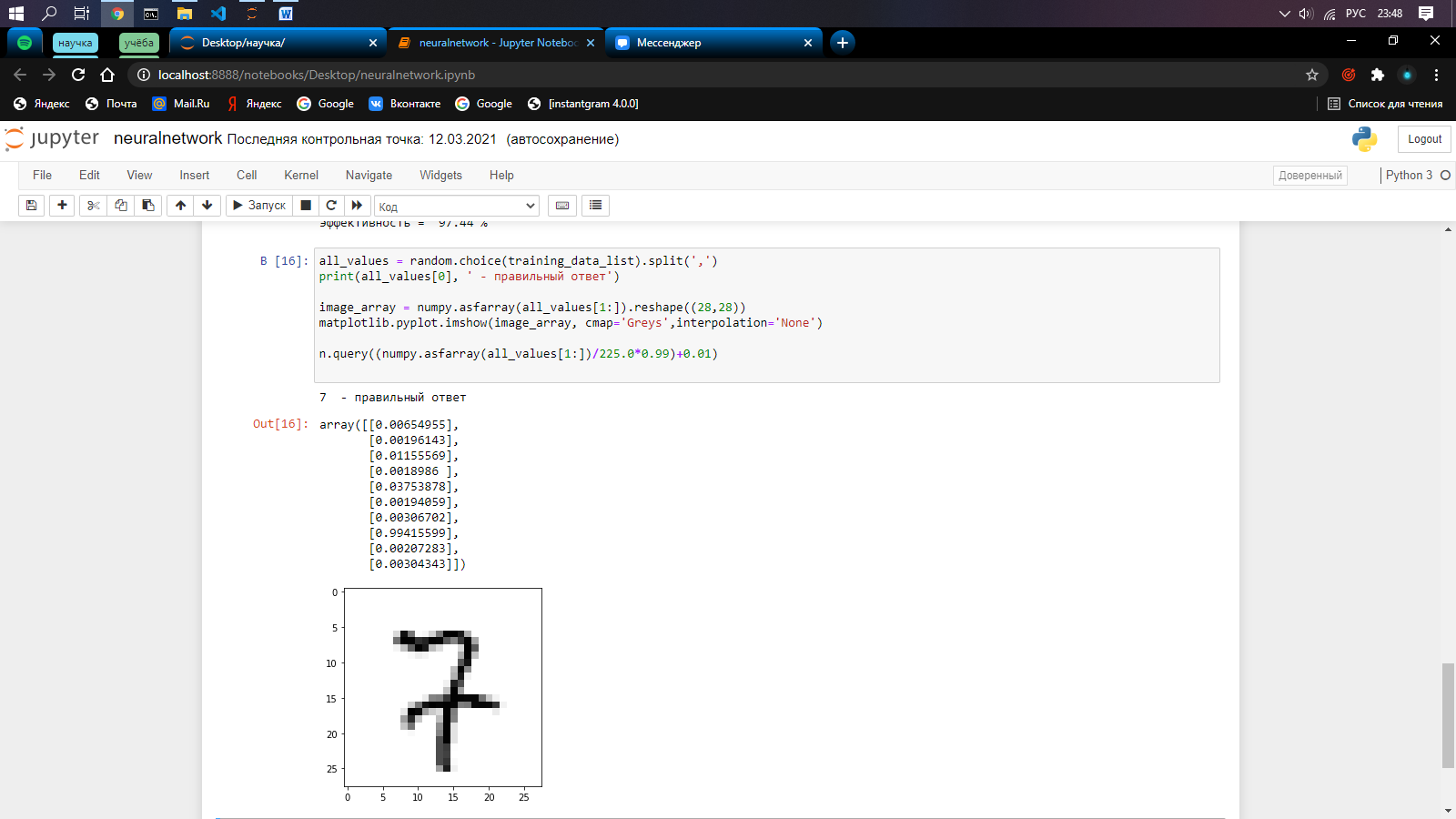
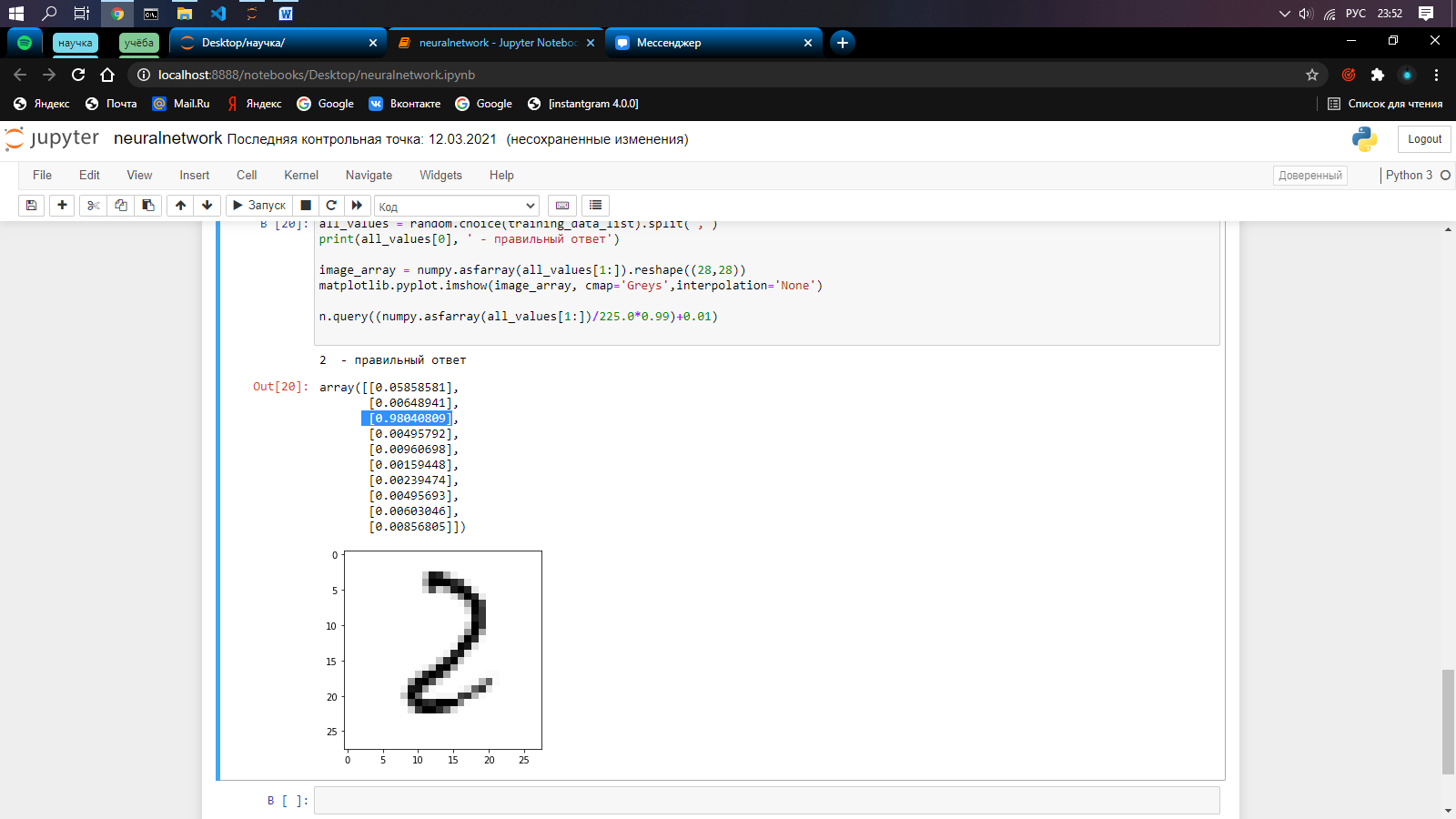


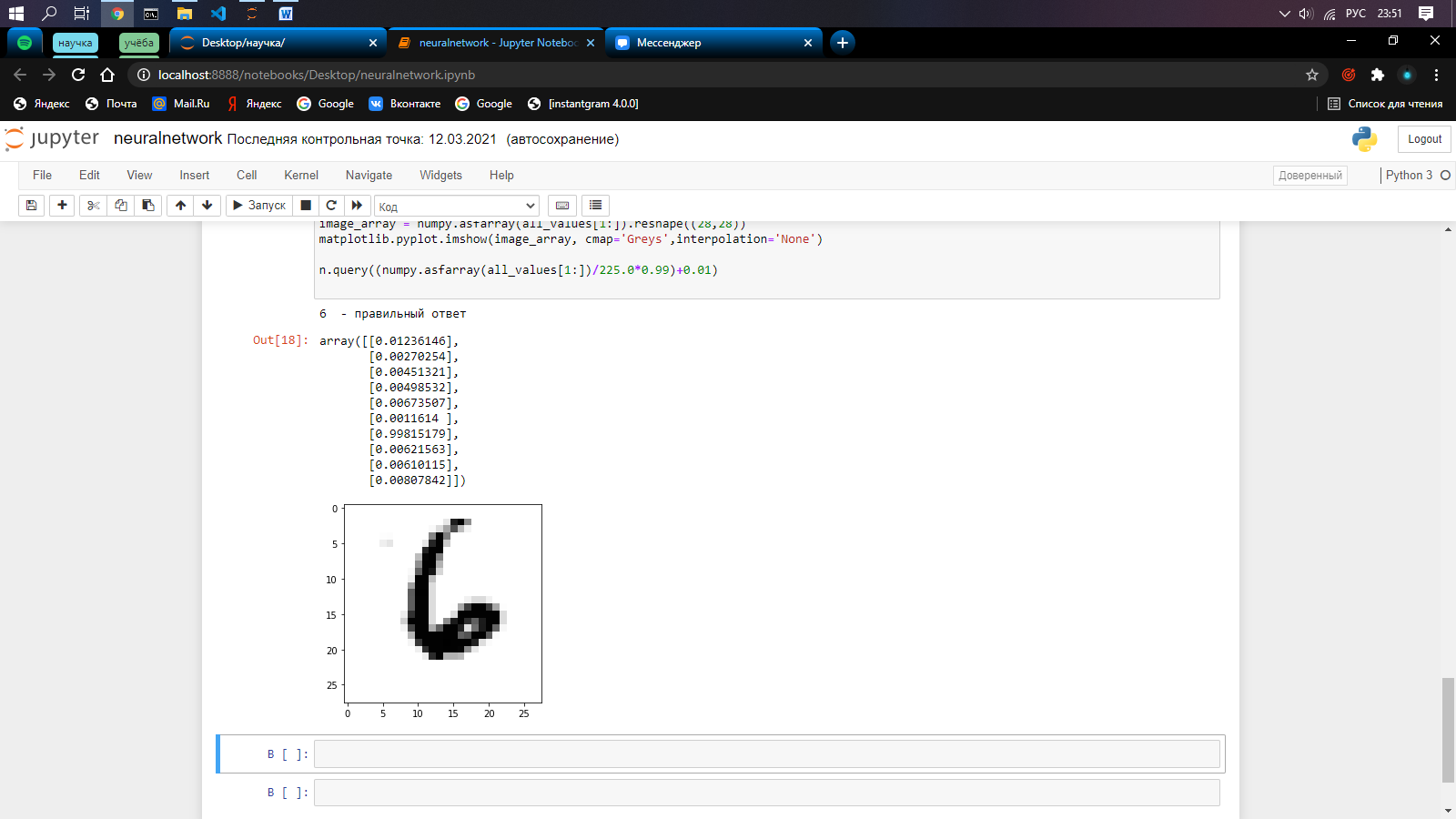










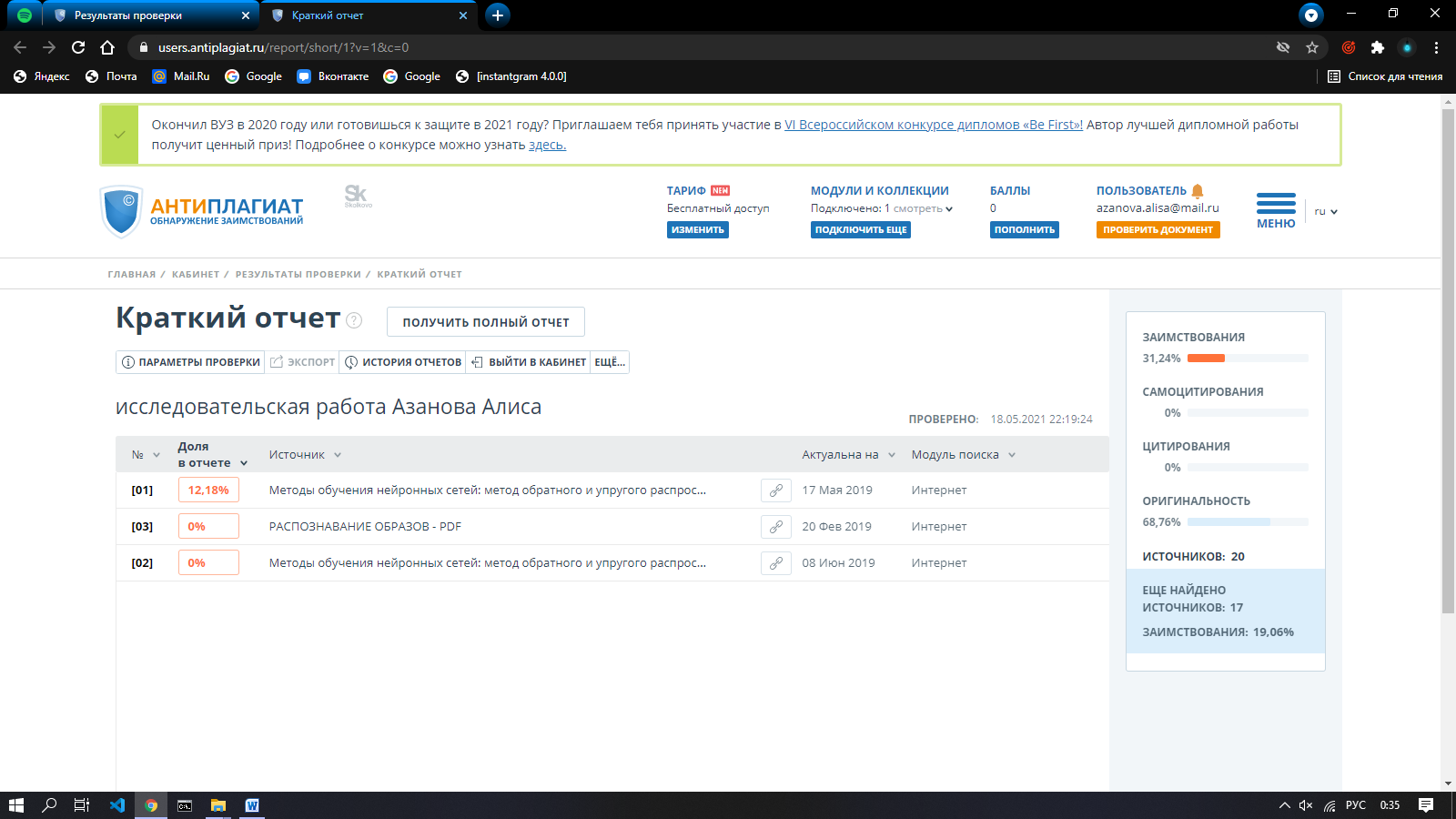


# **ЗАКЛЮЧЕНИЕ**

Безусловно, существует множество направлений по развитию данной программы. Возможно осуществить распознавание не только цифр, но и других символов, причём следующих подряд (т.е. распознавание чисел или текста), по аналогичному алгоритму, так же существует потенциал по изменению способов ввода символов, например, имитируя написание на бумаге текстуры и написание символов на данных текстурах различными предметами.   
 Так же, если на основе данной программы написать распознавание рукописного текста, и обучить нейронную сеть отличать подделку написанного от оригинала, то данную технологию можно использовать при экспертизе почерка.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Грокаем глубокое обучение – Эндрю Траск – издательство: Питер
2. Создаём нейронную сеть – Рашид Тарик – издательство: Вильямс
3. Стандартная библиотека Python 3. Справочник с примерами - Хеллман Даг - издательство: Вильямс

**ОТЧЕТ О ЗАИМСТВОВАНИЯХ (С САЙТА АНТИПЛАГИАТ.РУ)**