

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Федорова Анна Алексеевна

Выпускная квалификационная работа бакалавра

**Распознавание английского текста
сверточной нейронной сетью**

Направление 010400.62

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Коровкин М.В.

Санкт-Петербург

2016

Содержание.

Введение	3
Постановка задачи	6
Обзор литературы	7
Глава 1. Нейронные сети: основные понятия и применение	9
1.1. Отличие человеческого мозга от компьютера	9
1.2. Архитектура искусственной нейронной сети	10
1.3. Математическая модель искусственной нейронной сети	11
Глава 2. Модель сверточной нейронной сети	13
2.1. Архитектура сверточной нейронной сети	13
2.2. Математическая модель сверточной нейронной сети	15
2.2.1. Математическая модель сверточного слоя	15
2.2.2. Математическая модель субдискретизирующего слоя	17
2.2.3. Математическая модель выходного слоя	18
Глава 3. Обучение сверточной нейронной сети	19
3.1. Вычисление градиента функции ошибки сети	19
3.2. Метод обратного распространения ошибки	20
3.3. Математическая постановка задачи	21
Глава 4. Проведение эксперимента	22
4.1. Оборудование и программное обеспечение	22
4.2. Описание базы данных	22
4.3. Предварительная обработка изображений	22
4.4. Описание архитектуры реализованной нейронной сети	23
4.5. Сегментация текста на изображении	26
4.6. Результаты эксперимента	27
Выводы	28
Заключение	29
Список литературы	30

Введение.

В последнее десятилетие машинное обучение стало набирать все большую и большую популярность и сегодня занимает одну из лидирующих позиций в сфере информационных технологий. Объем цифровых данных, используемый нами в повседневной жизни, увеличивается с каждым днем, в связи с этим, возникла необходимость в автоматическом анализе смарт-данных для дальнейшего развития технологического прогресса.

Сегодня машинное обучение активно используется для распознавания речи и образов, а также на его основе построены многие поисковые системы. Машинное обучение настолько прочно вошло в нашу повседневную жизнь, что ежедневное использование его методов остается для нас, как правило, незамеченным.

Основной целью машинного обучения является обучение компьютера таким образом, чтобы он самостоятельно, без помощи человека, мог принимать решения, основанные на предыдущем опыте, для решения поставленной перед ним задачи. Существующие алгоритмы глубокого обучения помогают анализировать данные продаж в прошлом для предсказания поведения клиентов в будущем, оптимизировать поведение робота для минимизации используемых ресурсов и извлекать данные из биологических источников информации.

Активное развитие машинного обучения привело к распространению искусственных нейронных сетей [20]. Нейронные сети, основанные на биологической структуре человеческого мозга, превышают по своей вычислительной возможности прочие алгоритмы машинного обучения.

Искусственная нейронная сеть состоит из большого числа тесно связанных между собой элементов обработки информации, так называемых нейронов, работающих одновременно. Обучение нейронной сети схоже с обучением человека: основной идеей является запоминание образцов и примеров чего-либо. Нейронная сеть строится для решения конкретной задачи: распознавания образов или классификации данных.

Основным преимуществом использования нейронных сетей является возможность их применения для извлечения информации из больших или неточных данных, выявления основных признаков и тенденций, которые не могут быть распознаны человеком и прочими методами глубокого обучения. Обученная нейронная сеть может выступать в качестве «эксперта» для той категории данных, для которой она была подготовлена. Более того, подобного рода структура в дальнейшем может помочь прогнозировать новые результаты.

Одной из самых ярких и значительных форм искусственных нейронных сетей является сверточная нейронная сеть.

Сверточная нейронная сеть была впервые предложена Я.Лекуном и Й.Бенгуа [3]. Ученые исследовали зрительную кору головного мозга кошки, содержащую карты местных рецептивных полей, уменьшающиеся по мере детализации объекта, на который смотрит животное. Я.Лекун и Й.Бенгуа пришли к выводу, что математическую модель поведения зрительных рецепторов кошки можно построить, опираясь на следующие этапы обработки изображений:

- свертка исходного изображения объекта при помощи нескольких небольших фильтров;
- субдискретизация (объединение) полученных на предыдущем шаге ключевых признаков;
- повтор предыдущих шагов (свертки, а затем субдискретизации) до тех пор, пока на выходе не получится достаточное количество признаков исходного изображения;
- использование модели полносвязного слоя для получения решения конкретной задачи.

Таким образом, сверточная нейронная сеть благодаря своей архитектуре в основном используется для решения сложных графических задач распознавания и классификации изображений.

Сверточная нейронная сеть имеет ряд значительных преимуществ, благодаря которым она обходит по вычислительной мощности алгоритмы распознавания прошлых десятилетий [15].

В первую очередь, алгоритмы, использующие сверточные нейронные сети, инвариантны к различным искажениям, таким как поворот камеры или неравномерное распределение света на изображении, горизонтальным или вертикальным сдвигам и прочим. Также, сверточная нейронная сеть не требует выделения большого количества памяти для хранения извлеченных в процессе работы признаков. Еще одним преимуществом использования сверточной нейронной сети является достаточно быстрая скорость обучения, которая достигается благодаря уменьшению количества используемых параметров. Таким образом, производительность сверточной нейронной сети превышает в несколько раз производительность прочих нейронных сетей, используемых в задачах распознавания.

Постановка задачи.

Основная цель данной работы – показать, что алгоритмы, использующие сверточные нейронные сети могут быть применены к реально существующим проблемам классификации и идентификации. Для этого, рассмотрим задачу оффлайн-распознавания текста, то есть, распознавание образов с входного изображения в текстовый формат.

Обзор литературы.

Распознавание символов с изображения – это процесс, который ставит в соответствие каждому символу входного изображения некоторый заранее определенный класс. Рост интереса к различного рода компьютерным приложениям у современного человека вызвал острую необходимость научить компьютер «читать». Для этого были разработаны различные системы, каждая из которых должна состоять из следующих блоков [12]:

- предварительная обработка изображений;
- сегментация элементов текста;
- выделение основных признаков на изображении;
- распознавание символов с изображения;
- обработка результатов распознавания.

Наиболее популярными системами распознавания символов являются: классификаторы сопоставления, статистические классификаторы и искусственные нейронные сети.

Классификаторы сопоставления с некоторым шаблоном используются для упорядочивания неизвестных символов изображения по определенным классам [9]. Данный подход основан на минимизации расстояния между входным изображением и некоторым классом. В качестве индекса подобия в такой системе выступает расстояние: меньшее значение соответствует большей схожести. В частности, в некоторых алгоритмах мерой подобия служит Евклидово расстояние, вычисляемое между входным изображением и тестовым. Одним из главных недостатков данного метода является обработка каждого пикселя изображения в независимости от его принадлежности к какому-либо классу.

Статистические классификаторы основаны на вероятностном подходе [2]. Основной идеей таких алгоритмов является вычисление вероятности отношения к некоторому классу классифицируемого объекта. Так, объект будет относиться к тому классу, для которого вычисленная вероятность

оказалась наибольшей. Основным недостатком данного подхода является требование больших вычислительных мощностей.

Среди рассмотренных подходов для решения задачи распознавания символов с изображения наибольшую популярность имеют нейронные сети [5]. Для применения подобного рода систем используют предварительно созданную базу данных, а затем с помощью нее проводят обучение нейронной сети. Другими словами, нейронная сеть использует примеры обучающей выборки для дальнейшего распознавания символов с изображения. Более того, за счет увеличения размеров базы данных точность сети, как правило, увеличивается. Нейронные сети имеют ряд существенных преимуществ: способность обнаружения сложных нелинейных связей между зависимыми и независимыми переменными, минимальное использование статистических данных, а также существует возможность разработки сети с использованием нескольких алгоритмов обучения [8].

Глава 1.

Нейронные сети: основные понятия и применение.

1.1. Отличие человеческого мозга от компьютера.

Каждому из нас не раз доводилось слышать сравнение человеческого мозга с компьютером и, на первый взгляд, у них, действительно, есть нечто общее.

Человеческий мозг состоит из 100 миллионов клеток, получивших название – нейроны [4]. Основной функцией нейрона является обработка, получение и передача информации, поступающей из мозга в организм и наоборот. Каждый нейрон, как правило, состоит из тела, присоединенных к нему дендритов и аксона. Дендриты служат для сбора информации от других клеток и ее передачи в тело нейрона. Аксон, напротив, поставляет информацию от рассматриваемого тела клетки к остальным нейронам (рис.1).

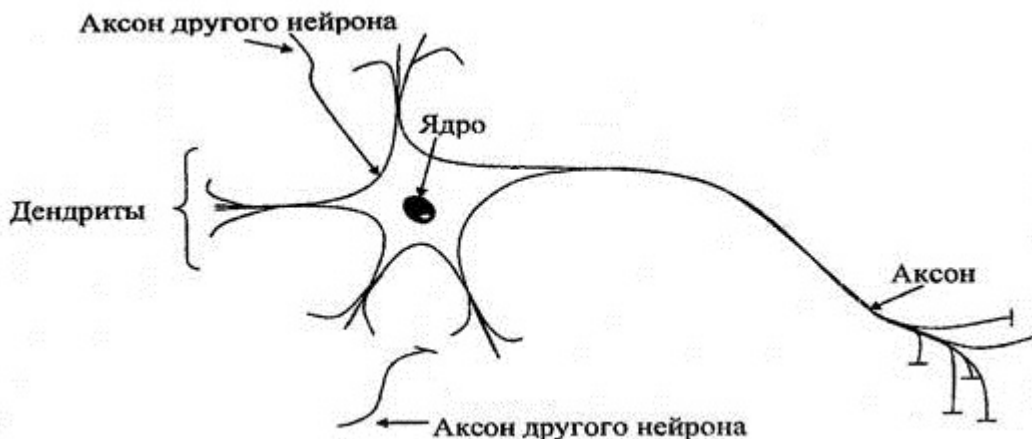


Рис.1. Модель нейрона человеческого мозга.

Эквивалентом клеток человеческого мозга в устройстве компьютера служат транзисторы – миниатюрные переключатели сигнала. Подобно нейронам, они изменяют выходное значение в зависимости от входных данных. Количество транзисторов в процессоре каждого компьютера различно: от 50 миллионов до 2 миллиардов.

Пожалуй, это единственное сходство, которое можно найти, анализируя работу человеческого мозга и компьютера. Основным различием является по-

разному организованная способность к «мышлению». Если в устройстве компьютера каждый транзистор соединен последовательно с еще одним или двумя транзисторами, то в человеческом мозге каждый нейрон соединен параллельно с более чем 10000 своих соседей, что увеличивает его мощность восприятия информации.

1.2. Архитектура искусственной нейронной сети.

Основная идея, воплощенная в архитектуре нейронной сети, заключается в имитации плотно соединенных между собой клеток головного мозга при помощи компьютера. При этом, полученная система должна быть достаточно легко обучаемая для распознавания образов или принятия каких-либо решений, подобно тому, как эти действия осуществляются человеком. Одним из главных достоинств искусственных нейронных сетей является их возможность действовать согласно предыдущему опыту, не требуя дополнительного программирования, что делает их схожими с устройством человеческого мозга [14].

Искусственная нейронная сеть состоит из десятков, тысяч, иногда даже миллионов нейронов, объединенных в слои. Так, входной слой состоит из нейронов, предназначенных для принятия различных форм информации от внешнего мира, которая в дальнейшем будет обрабатываться нейронной сетью. С противоположной ему стороны расположен слой выходных нейронов, реагирующих на поступающую информацию. Между рассмотренными слоями располагается несколько промежуточных слоев, которые содержат нейроны, обрабатывающие информацию.

Нейроны между слоями сети соединены с помощью специальных связей, именуемых весами, которые могут быть как положительными (в случае, если предыдущий нейрон возбуждает следующий за ним), так и отрицательными (если предыдущий нейрон тормозит следующий за ним).

1.3. Математическая модель искусственной нейронной сети.

Далее, под искусственной нейронной сетью будем понимать упорядоченную тройку элементов (N, V, w) [6]. Здесь, N – это множество нейронов сети, $V = \{(i, j) | i, j \in N\}$ – множество, элементы которого представляют связи между нейроном i и j . $w: V \rightarrow \mathbb{R}$, $w(i, j)$ или $w_{i,j}$ – весовая функция между нейронами i и j .

При рассмотрении конкретного нейрона j , видно, что к данному нейрону присоединено достаточно большое количество связей, идущих от остальных нейронов сети, то есть, передающих сигнал нейрону j . К нейрону j функция активации доставляет выходы $o_{i_1}, o_{i_2}, \dots, o_{i_n}$ нейронов i_1, i_2, \dots, i_n , имеющих связь с j , и присоединяет к ним соответствующие веса $w_{i,j}$, тем самым формируя вход нейрона net_j . Таким образом, вход нейрона – это результат некоторой функции.

Из представленных выше рассуждений, можем сформулировать определение входа конкретного нейрона сети.

Пусть $I = \{i_1, i_2, \dots, i_n\}$ – множество нейронов, таких, что $\forall z \in \{1, \dots, n\}: \exists w_{i_z, j}$. Тогда, вход нейрона net_j вычисляется как (рис.2):

$$net_j = f_{prop}(o_{i_1}, o_{i_2}, \dots, o_{i_n}, w_{i_1}, w_{i_2}, \dots, w_{i_n}).$$

Часто в качестве функции f_{prop} используют взвешенную сумму вида:

$$net_j = \sum_{i \in I} (o_i \cdot w_{i,j}).$$

Реакцию каждого нейрона на поступивший сигнал будем определять с помощью функции активации, а под состоянием активации (возбужденности) a_j нейрона j будем понимать результат данной функции. Также рассмотрим пороговое значение b_j , которое определяется для каждого конкретного нейрона j , как максимальное значение градиента функции активации. С биологической точки зрения, пороговое значение представляет собой величину, при которой нейрон начинает возбуждаться.

Таким образом, функция активации нейрона j имеет вид (рис.2):

$$a_j = f_{act}(net_j, a_{j-1}, b_j).$$

Выходная функция нейрона j вычисляет значения, передаваемые на вход остальным нейронам, присоединенным к j . С математической точки зрения выход нейрона o_j представляет собой результат воздействия некоторой функции f_{out} (рис.2):

$$f_{out}(a_j) = o_j.$$

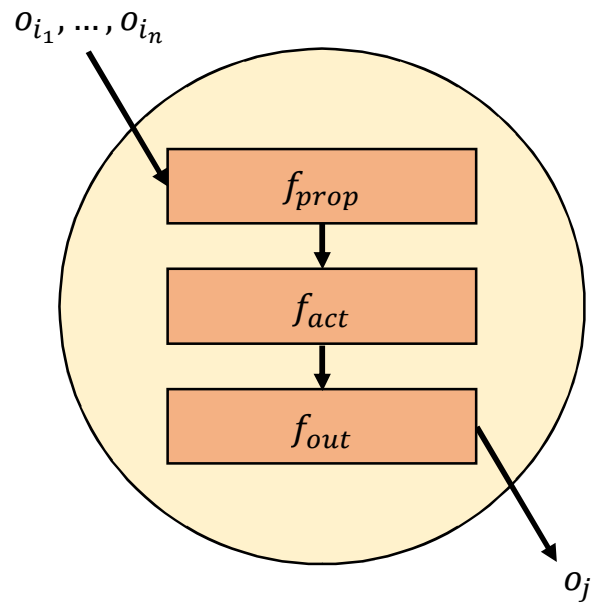


Рис.2. Обработка данных нейроном j .

Глава 2.

Модель сверточной нейронной сети.

2.1. Архитектура сверточной нейронной сети.

Сверточные нейронные сети (СНС) являются одним из самых важных классов теории глубокого машинного обучения для решения задач компьютерного зрения. СНС состоят из нескольких, как правило, чередующихся слоев обработки, каждый из которых содержит, как линейные, так и нелинейные операторы. Сегодня использование СНС является одним из основных методов для извлечения признаков из аудио, видео и текстовых данных.

СНС состоит из трех основных видов слоев: сверточный слой, субдискретизирующий слой и выходной слой (чаще всего полносвязный) [10]. Слои СНС расположены друг за другом: сначала сверточный слой, а затем субдискретизирующий, за последним сверточным слоем следует выходной слой (рис.3). Сверточный и субдискретизирующий слои считаются слоями двумерной размерности, а выходной слой, как правило, представляет собой вектор из пространства \mathbb{R}^1 . В СНС каждый двумерный слой имеет несколько уровней. Каждый уровень представляет собой двумерный массив. Выход каждого уровня в дальнейшем будем называть картой признаков.

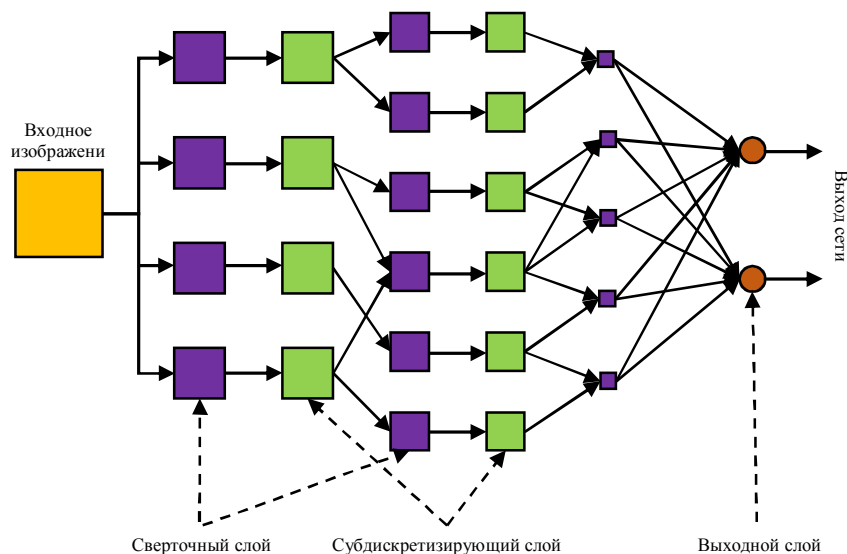


Рис.3. Архитектура сверточной нейронной сети.

Основной идеей использования сверточного слоя является применение математической операции свертки (фильтра) к изображению [19]. Свертка – это двумерная матрица коэффициентов. Вход такого фильтра – это фрагмент двумерного изображения, а выход – некоторое число (рис.4). Преимущество использования подобного рода фильтров заключается в следующем: число на выходе тем больше, чем больше элемент изображения похож на применяемый к нему фильтр. Следовательно, использование операции свертки помогает получить на выходе изображение, каждый пиксель которого будет соответствовать степени подобия кусочка изображения на фильтр. Иными словами, мы получим карту признаков.

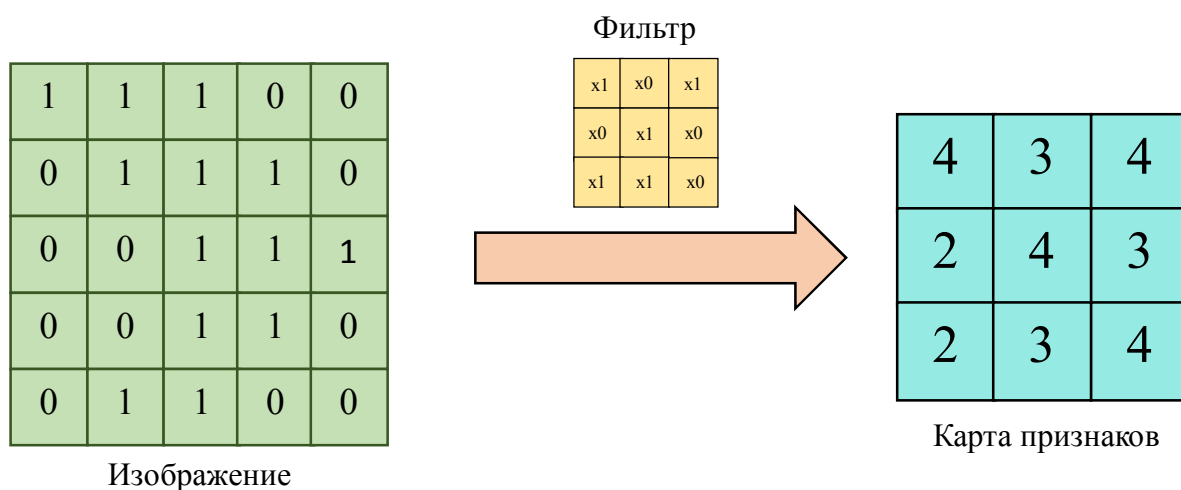


Рис.4. Пример использования двумерной операции свертки для формирования карты признаков.

Субдискретизация применяется для уменьшения общего размера изображения и увеличения степени инвариантности применяемых к нему фильтров. При рассмотрении архитектуры СНС будем опираться на факт, что наличие какого-либо признака на изображении гораздо важнее, чем точное знание его координат. Таким образом, сутью субдискретизации является выбор максимального нейрона из нескольких соседних. Затем, этот нейрон принимаем за элемент последующей, но уже уменьшенной карты признаков. Данная операция помогает увеличить инвариантность к масштабу входного изображения.

Таким образом, СНС строится посредством чередования слоев свертки и субдискретизации. Подобная архитектура нейронной сети позволяет строить карты признаков из карт признаков, а это означает, что мы получили возможность распознавать иерархии признаков. Зачастую, на практике, в конце работы СНС карты признаков преобразуются в вектор или скаляр. Отсюда возникла потребность в добавлении полносвязного слоя на выходе.

Полносвязный слой является самым простым и широко применяемым слоем нейронной сети. Каждый нейрон в этом слое – персептрон с нелинейной функцией активации. В качестве функции активации принято использовать либо логистическую функцию вида: $f(x) = \frac{1}{1+e^{-x}}$, либо гиперболический тангенс: $f(x) = A \cdot \tanh(Bx)$, где $A, B = const$. Каждый нейрон полносвязного слоя соединяется с выходами нейронов предыдущего слоя, суммирует свои входы, умножает на веса, добавляет порог и результат подает на вход функции активации. Полученное значение является выходом нейрона.

2.2. Математическая модель сверточной нейронной сети.

Для описания математической модели рассматриваемой нейронной сети будем использовать следующие обозначения.

Под $l \in [1; L]$ будем понимать рассматриваемый в данный момент слой нейронной сети, где $L = 2a + 2, a \in \mathbb{Z}^+$ – количество слоев в сети. За N^l обозначим количество карт признаков на слое l , а за $f_l(\cdot)$ – функцию активации рассматриваемого слоя l . Также, под переменной y_n^l будем понимать n -ую карту признаков на слое l [17].

2.2.1. Математическая модель сверточного слоя.

Введем в рассмотрение сверточный слой l . В подобной архитектуре нейронной сети l принимается нечетным числом, то есть $l = 1, 3, \dots, 2a + 1$. Тогда, для карты признаков n будет иметь место следующее:

- $\mathbb{w}_{m,n}^l = \{w_{m,n}^l(i, j)\}$ – свертка, применяемая к карте признаков m слоя $(l - 1)$, на слое l с картой признаков n ;
- b_n^l – пороговые значения, присоединяемые к карте признаков n на слое l ;
- V_n^l – список всех уровней слоя $(l - 1)$, которые соединяются с картой признаков n слоя l .

Таким образом, карта признаков n сверточного слоя l будет вычисляться следующим образом:

$$\psi_n^l = f_l(\sum_{m \in V_n^l} \psi_m^{l-1} \otimes \mathbb{w}_{m,n}^l + b_n^l),$$

где под оператором \otimes понимается математическая операция двумерной свертки.

Предположим, что размер входных карт признаков ψ_m^{l-1} равен $H^{l-1} \times W^{l-1}$, а размер применяемой к ним свертки $\mathbb{w}_{m,n}^l$ равняется $r^l \times c^l$, тогда размер выходной карты признаков ψ_n^l вычисляется как (рис.5):

$$(H^{l-1} - r^l + 1) \times (W^{l-1} - c^l + 1).$$

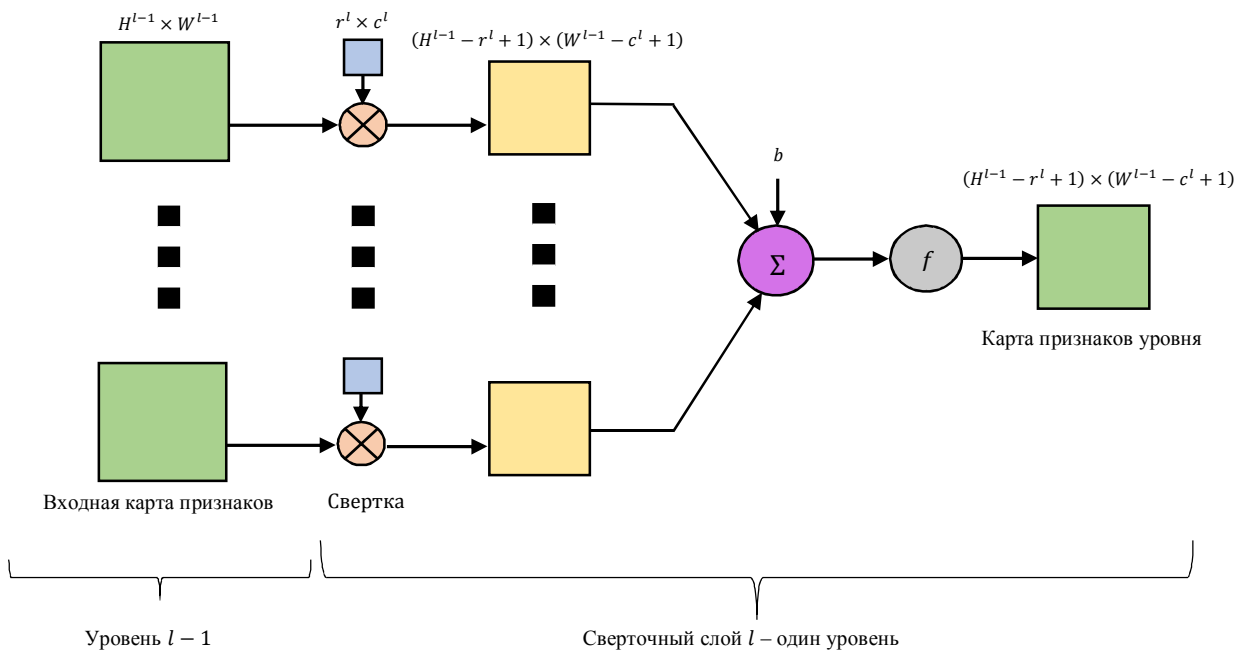


Рис.5. Схема сверточного слоя l .

2.2.2. Математическая модель субдискретизирующего слоя.

Введем в рассмотрение субдискретизирующий слой l . В сверточной нейронной сети l принято принимать четным числом, то есть $l = 2, 4, \dots, 2a$. Для карты признаков n введем следующие обозначения: $w_{m,n}^k$ – фильтр, применяемый к n на слое l , и b_n^l – добавочное пороговое значение.

Далее будем действовать следующим образом: разделим карту признаков n $(l - 1)$ -ого слоя на непересекающиеся блоки размером 2×2 пикселя. Затем просуммируем значения четырех пикселей в каждом блоке и в результате получим матрицу $z_n^{l-1} = \{z_n^{l-1}(i, j)\}$, элементами которой будут являться соответствующие значения сумм. Таким образом, формула для вычисления значений элементов матрицы будет иметь следующий вид:

$$z_n^{l-1} = y_n^{l-1}(2i - 1, 2j - 1) + y_n^{l-1}(2i - 1, 2j) + y_n^{l-1}(2i, 2j - 1) + y_n^{l-1}(2i, 2j).$$

Карта признаков n субдискретизирующего слоя l вычисляется, как:

$$y_n^l = f_l(z_n^{l-1} \times w_{m,n}^l + b_n^l).$$

Благодаря представленным выше рассуждениям, становится возможным посчитать размер $H^l \times W^l$ карты признаков y_n^l субдискретизирующего слоя l (рис.6):

$$H^l = \frac{H^{l-1}}{2}, W^l = \frac{W^{l-1}}{2}.$$

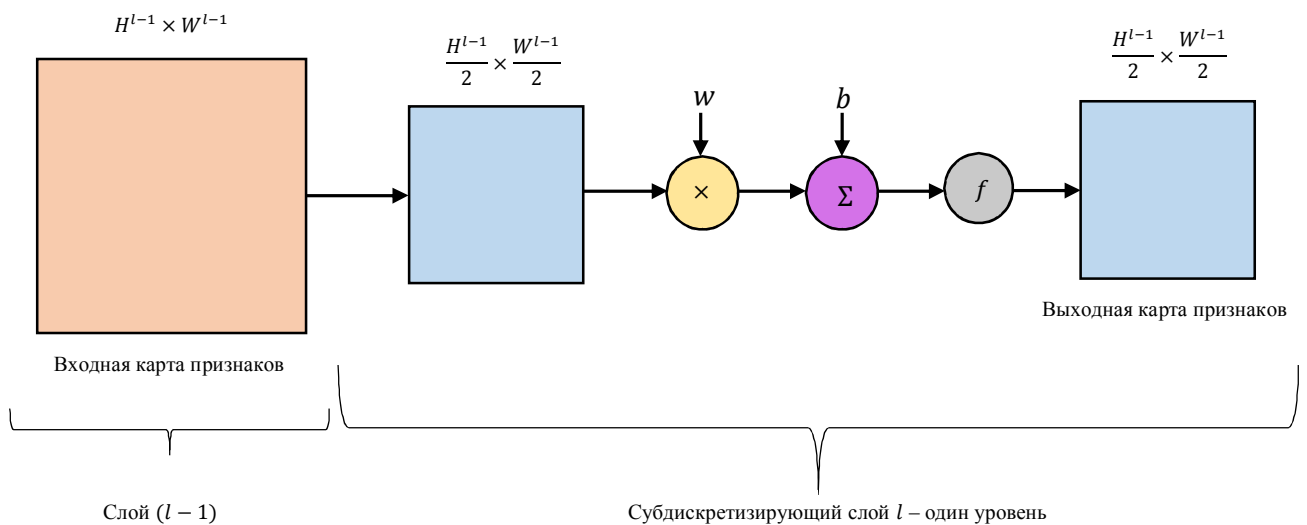


Рис.6. Схема субдискретизирующего слоя l .

2.2.3. Математическая модель выходного слоя.

В данном параграфе будем рассматривать выходной слой L , состоящий из единичных нейронов. Примем за N^L — количество нейронов на данном слое. Как и при рассмотрении предыдущих слоев, обозначим за $w_{m,n}^L$ — фильтр, применяемый к карте признаков m последнего сверточного слоя для получения перехода к нейрону n выходного слоя. Пусть b_n^L — пороговое значение, добавляемое к нейрону n .

Пользуясь введенными обозначениями, получаем формулу для подсчета значения выходного нейрона n :

$$y_n^L = f_L(\sum_{m=1}^{N^{L-1}} y_m^{L-1} w_{m,n}^L + b_n^L).$$

Таким образом, выходом сверточной нейронной сети является вектор следующего вида:

$$y = [y_1^L, y_2^L, \dots, y_{N^L}^L].$$

Глава 3.

Обучение сверточной нейронной сети.

Основной целью обучения нейронных сетей, предназначенных для решения различных задач распознавания, является минимизация функции ошибки в течение нескольких итераций (эпох).

Далее будем предполагать, что обучающее множество содержит P изображений. Обозначим за Z^p – p -ое изображение тренировочного множества, а за d_n^p – желаемый p -ый выход системы. Тогда, функция ошибки сети определяется как [7]:

$$E(\mathbf{w}) = -d_n^p \cdot \log \psi_n^p, \quad (1)$$

где ψ_n^p – выход нейронной сети.

3.1. Вычисление градиента функции ошибки сети.

Для минимизации рассмотренной выше функции ошибки (1) необходимо вычислить ее градиент, что требует знания частных производных по основным параметрам сети.

Таким образом, для каждого нейрона (i, j) карты признаков n слоя l , $l = 1, \dots, L$, частная производная по взвешенной сумме $s_n^{l,p}(i, j)$, где $n = 1, \dots, N^l$, $i = 1, \dots, H^l$, $j = 1, \dots, W^l$ имеет вид:

$$\delta_n^{l,p}(i, j) = \frac{\partial E}{\partial s_n^{l,p}(i, j)}. \quad (2)$$

Используя цепное правило дифференцирования, получаем возможность выразить производную (2) через функцию активации слоя.

Для выходного слоя L (2) будет иметь вид:

$$\delta_n^{l,k} = \frac{1}{P \times N^l} \cdot e_n^p \cdot f'(s_n^{l,p}), \quad (3)$$

где $e_n^p = \psi_n^{L,p} - d_n^p$.

Для сверточного слоя $l = 2a + 1$ (2) будет иметь вид:

$$\delta_n^{l,p}(i, j) = f_l'[s_n^{l,p}(i, j)] \times \delta_n^{l+1}(i, j) \times \mathbf{w}_{m,n}^{l+1,p}. \quad (4)$$

Для субдискретизирующего слоя $l = 2a$ (2) будет иметь вид:

$$\delta_n^{l,p} = f_l' [s_n^{l,p}(i,j)] \times \sum_{m \in U_n^l} \sum_{(i',j') \in R^l(i,j)} \delta_m^{l+1,p}(i',j') \cdot w_{m,n}^{l+1}(i-i', j-j'), \quad (5)$$

где $U_n^l = \{1, 2, \dots, N^{l-2}\}$, $R^l(i,j) = \{i = 1, 2, \dots, H^l; j = 1, 2, \dots, W^l\}$, $i' = \lfloor \frac{i}{2} \rfloor$, $j' = \lfloor \frac{j}{2} \rfloor$.

Используя вычисленные производные (3), (4), (5), получаем возможность вычислить градиент функции ошибки сети (1) для каждого слоя по параметрам $w_{m,n}^l$ и b_n^l .

Градиент (1) для выходного слоя L :

$$\frac{\partial E}{\partial w_{m,n}^l} = \sum_{p=1}^P \delta_n^{l,p} \psi_m^{l-1,p}; \quad \frac{\partial E}{\partial b_n^l} = \sum_{p=1}^P \delta_n^{l,p}, \quad (6)$$

где $m = 1, 2, \dots, N^{l-1}$, $n = 1, 2, \dots, N^l$.

Градиент (1) для сверточного слоя $l = 2a + 1$:

$$\frac{\partial E}{\partial w_{m,n}^l} = \sum_{p=1}^P \delta_n^{l,p}(i,j) \cdot \psi_n^{l-1,p}; \quad \frac{\partial E}{\partial b_n^l} = \sum_{p=1}^P \sum_{(i,j)} \delta_n^{l,p}(i,j), \quad (7)$$

где $m = 1, 2, \dots, N^{l-1}$, $n = 1, 2, \dots, N^l$.

Градиент (1) для субдискретизирующего слоя $l = 2a$:

$$\begin{aligned} \frac{\partial E}{\partial w_{m,n}^l} &= \sum_{p=1}^P \sum_{(i',j')} \delta_n^{l,p}(i',j') \cdot \psi_n^{l-1,p}(i',j'); \\ \frac{\partial E}{\partial b_n^l} &= \sum_{p=1}^P \sum_{(i',j')} \delta_n^{l,p}(i',j'), \end{aligned} \quad (8)$$

где $m = 1, 2, \dots, N^{l-1}$, $n = 1, 2, \dots, N^l$, $i' = \lfloor \frac{i}{2} \rfloor$, $j' = \lfloor \frac{j}{2} \rfloor$.

3.2. Метод обратного распространения ошибки.

В основе метода лежит стохастический градиентный спуск: настройка параметров системы для достижения минимума функции ошибки производится после одного случайно выбранного обучающего образца, предъявленного сети [11]. То есть, обновление весов осуществляется сразу после вычисления градиента для текущего примера. Таким образом, минимизация осуществляется в направлении антиградиента:

$$\Delta w = -\alpha \cdot \nabla E^p,$$

$\alpha > 0$ – скорость обучения сети.

Пользуясь выведенными в пункте 3.1. значениями производных, распишем алгоритм подробнее.

На первом шаге задаем произвольные значения весовых коэффициентов. Затем, передаем сигнал в сети в прямом направлении: от предыдущих слоев к последующим. На этом же шаге вычисляем значение функции ошибки для выходного слоя.

Вторым шагом обучения сети является распространение ошибки, полученной на первом этапе, в обратном направлении, то есть от последнего слоя к предыдущим. При этом происходит корректировка весов и пороговых значений каждого входа нейрона с использованием градиентного спуска. То есть, вычисляем новые значения весов и порогов:

$$w' = w + \alpha \cdot \frac{\partial E}{\partial w}; \quad b' = b + \alpha \cdot \frac{\partial E}{\partial b},$$

где $\frac{\partial E}{\partial w}, \frac{\partial E}{\partial b}$ вычисляются по формулам (6), (7), (8).

3.3. Математическая постановка задачи.

Таким образом, цель данной работы – минимизировать функцию потерь $E(w)$ за счет выбора настраиваемых параметров обучаемой системы w . $E(w)$ представляет собой арифметическое среднее функций вида: $E^p = D(D^p, F(w, Z^p))$, где Z^p – p -ый вход системы, D^p – несоответствие между желаемым выходом системы для Z^p и значением, которое выдает система. $E(w)$ будем вычислять над обучающим множеством пар вида: $\{(Z^1, D^1), \dots, (Z^p, D^p)\}$.

Глава 4.

Проведение эксперимента.

4.1. Оборудование и программное обеспечение.

Эксперимент был осуществлен на ПК с процессором Intel Core i3-2367M (1.40 GHz). Разработка модели сверточной нейронной сети осуществлялась в Matlab с использованием MATLAB Library for Convolutional Neural Networks [17].

4.2. Описание базы данных.

Для тестирования разработанной системы была использована база данных, содержащая различные начертания английских букв: a, b, \dots, z , и соответствующие им метки. Для обучения сети было использовано 24206 изображений, из которых 15106 изображений использовалось в качестве обучающей выборки и 9100 изображений в качестве тестовой. Размер каждого изображения составляет 32×32 пикселя. Каждый пиксель кодируется числом в интервале $[0; 1]$, где 0 соответствует черному цвету, а 1 – белому (рис.7).



Рис.7. Пример 10 изображений обучающей выборки для буквы k .

4.3. Предварительная обработка изображений.

Предварительная обработка изображений необходима для достижения максимальной точности распознавания системы. Таким образом, полученная нейронная сеть будет инвариантна к незначительным искажениям: шуму, повороту и масштабированию.

Простые искажения изображения, такие как сдвиг, поворот и угловое смещение могут быть устранены путем применения простых аффинных преобразований. Также, для улучшения качества работы нейронной сети

подаваемые на вход изображения дополнительно усреднялись.

Данный подход позволил получить сеть, обучение которой осуществлялось с применением различного набора изображений на каждой эпохе.

4.4. Описание архитектуры реализованной нейронной сети.

В работе была реализована и использована модель сверточной нейронной сети, содержащая 7 слоев. Обучение сети производилось в течение 150 эпох.

Первый слой использованной нейронной сети – сверточный. Ему на вход подается изображение размером 32×32 пикселя, к которому применяется свертка размером 5×5 . Количество использованных фильтров – 20 (рис.8).

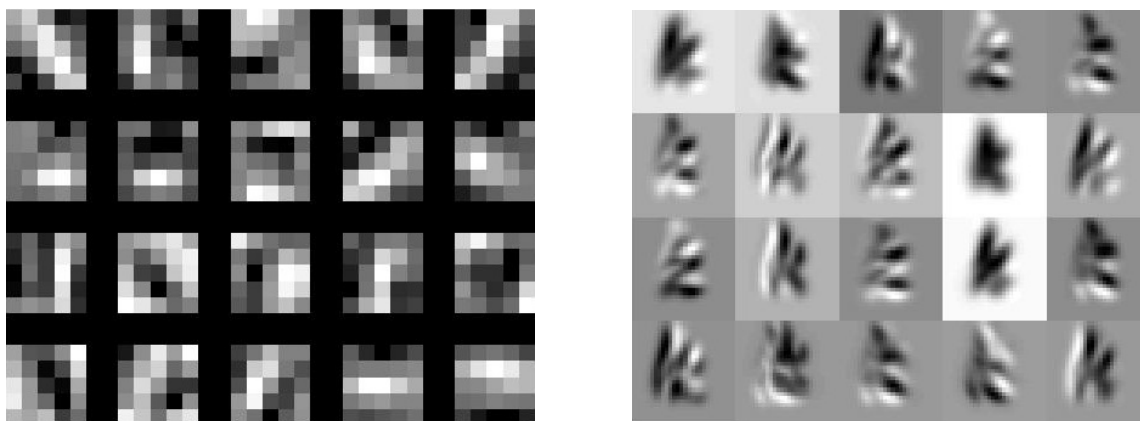


Рис.8. Фильтры первого слоя сети, настроенные на распознавание простейших признаков.

Размер выходных карт признаков данного слоя составляет 28×28 пикселей.

Второй слой реализованной сети – слой субдискретизации. Основной функцией данного слоя являлось уменьшение размерности входного изображения и выбор пикселя с максимальным значением среди нескольких соседних. В данной архитектуре сети такой выбор осуществлялся между 4 соседними пикселями (рис.9).

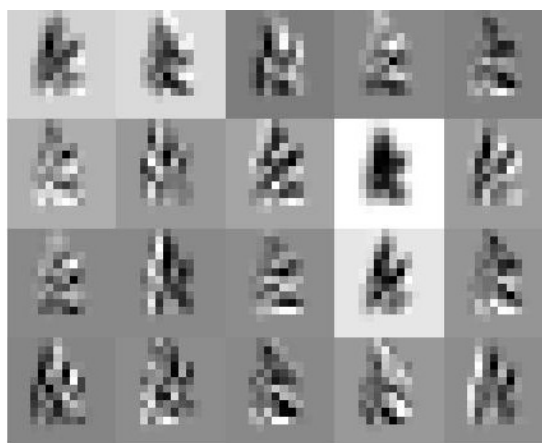


Рис.9. Применение субдискретизации к одному изображению обучающей выборки.

Третьим слоем рассматриваемой нейронной сети являлся сверточный слой, принимающий на вход 20 карт признаков предыдущего слоя. На данном этапе в качестве фильтров выбирались матрицы размера 5×5 , формируя на выходе 50 карт признаков размером 10×10 пикселей.

Четвертый слой, как и второй слой нейронной сети, являлся слоем субдискретизации. Выходом данного слоя послужили 50 карт признаков – 5×5 пикселей.

Пятый слой реализованной нейронной сети – сверточный слой. На вход данному слою подавалось 50 карт признаков предыдущего слоя, к которым применялись фильтры размера 4×4 . Таким образом, выход данного слоя представляет собой 500 карт признаков размером 2×2 пикселя.

Задачей шестого слоя было применение нелинейной функции вида [20]:

$$f(y_n^5) = \max(0, y_n^5(i, j)).$$

Использование данной функции наиболее удобно при обучении сверточной нейронной сети в силу простоты вычисления ее градиента (9), что значительно уменьшает время обучения сети, увеличивая при этом нелинейность выхода.

$$\frac{\partial f}{\partial y_n^5} = \begin{cases} 1, & y_n^5(i, j) > 0 \\ 0, & y_n^5(i, j) \leq 0 \end{cases} \quad (9)$$

Седьмой слой рассматриваемой нейронной сети – полносвязный слой, содержащий 26 нейронов. Основной задачей данного слоя являлась классификация. В качестве функции активации данного слоя выбиралась softmax-функция вида [15]:

$$\psi_{26}^7 = \frac{e^{y_n^6(i,j)}}{\sum_{k=0}^{25} e^{y_k^6(i,j)}}$$

а для сравнения получившихся значений функция:

$$E^7 = -d_{26}^7 \cdot \log \psi_{26}^7,$$

где d_{26}^7 – желаемый выход для конкретного обучающего примера.

Примеры работы реализованной сверточной нейронной сети представлены на рисунках 10, 11, 12.



Рис.10. Пример работы нейронной сети для произвольных букв (черные буквы – исходные изображения, синие – результат работы сети).

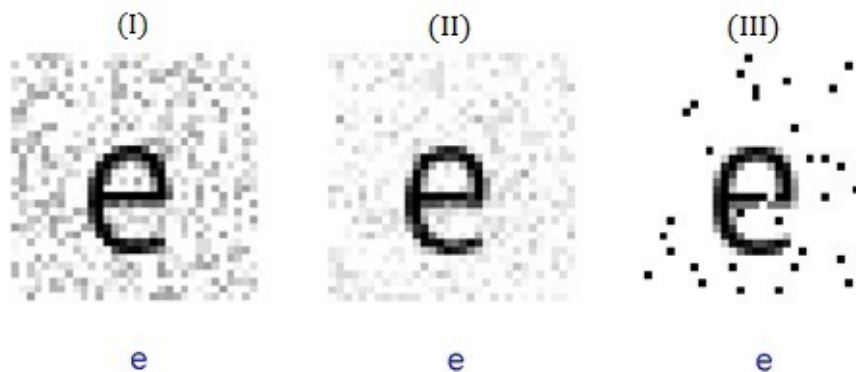


Рис.11. Пример работы нейронной сети для одной буквы с добавлением различного шума: мультипликативный шум (I), гауссовский шум (II), шум типа «соль и перец» (III) (черные буквы – исходные изображения, синие – результат работы сети).



Рис.12. Пример работы нейронной сети для одной буквы с искажениями (черные буквы – исходные изображения, синие – результат работы сети).

4.5. Сегментация текста на изображении.

Для использования полученной нейронной сети для распознавания подаваемых на вход слов и предложений с изображения необходимо осуществить предварительную сегментацию.

Основной идеей использованного алгоритма являлось нахождение всевозможных способов разделения букв на изображении, а затем использование полученной нейронной сети для отыскания наиболее правильного разреза [18].

В первую очередь было сгенерировано множество всевозможных разрезов, к ним относились столбцы вертикальных пикселей изображения, сумма которых была наименьшей. При использовании данного подхода будем делать предположение, что необходимый нам разрез находится в рассматриваемом множестве.

Затем строился направленный ациклический граф описанным далее способом. Каждая вершина графа соответствовала одному из полученных ранее разрезов, а ребро – фрагменту изображения, расположенному между разрезами на узлах. Ребро графа строилось только в том случае, если полученный фрагмент мог быть распознан в качестве буквы. К полученным фрагментам изображения применялась нейронная сеть, которая присваивала каждому фрагменту соответствующую букву. Пары последовательных фрагментов объединялись в буквы тогда и только тогда, когда результаты сети совпадали (рис.13).



Рис.13. Пример работы нейронной сети для одного предложения различного начертания (черные буквы – исходные изображения, синие – результат работы сети).

4.6. Результаты эксперимента.

Точность распознавания на тестовом множестве реализованной и использованной в данной работе модели сверточной нейронной сети составила 94,5% (рис.14,15).

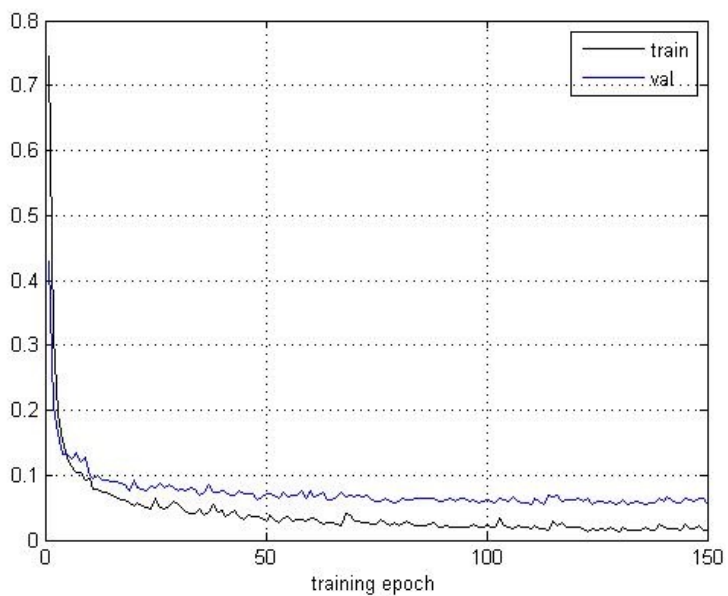


Рис.14. График зависимости точности от количества эпох нейронной сети.



Рис.15. Примеры некорректной работы нейронной сети.

Желаемые выходы слева направо: «d», «z», «p»
(черные буквы — исходные изображения, синие — результат работы сети).

Выводы.

Полученный алгоритм, использующий сверточную нейронную сеть, может быть применен для решения задач оффлайн-распознавания, так как обладает достаточно высокой точностью и инвариантностью к различным искажениям и шумам. Однако, эффективность реализованной системы зависит от размера обучающей и тестовой выборок.

Заключение.

Точность решения поставленной задачи с использованием модели сверточной нейронной может быть улучшена практически до 100%. Об этом подробнее написано в публикации [22].

В последние годы модели искусственных нейронных сетей, используемые для решения задач оффлайн-распознавания текста, приобрели огромную популярность. Наряду со сверточными нейронными сетями активно применяются и внедряются многомерные рекуррентные и, так называемые, deconvolutional нейронные сети.

Основным отличием deconvolutional сети от сверточной является суммирование по картам признаков, учитывая фильтры. Использование такого подхода позволяет получить свертку, применимую ко всему изображению целиком, а не к отдельной его части [13].

Многомерная рекуррентная нейронная сеть построена по принципу циклического графа: сигнал передается как от входного слоя к выходному, так и в обратном направлении. Поведение нейронов данной сети задается с помощью дифференциальных или разностных уравнений. Подобная архитектура делает сеть инвариантной к различного рода искажениям входных данных. Преимущества использования моделей многомерных рекуррентных сетей для решения задач оффлайн-распознавания текста более подробно описаны в [1].

Список литературы.

1. Alex Graves, Jurgen Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks // NIPS 2008, Vancouver, Canada. P.545-552.
2. Alex Smola, S.V.N.Vishwanatham. Introduction to Machine Learning // Cambridge University Press, 2008. P.20-32.
3. An introduction to Convolutional Neural Networks.
http://white.stanford.edu/teach/index.php/An_Introduction_to_Convolutional_Neural_Networks
4. An introduction to Neural Networks.
<http://www.explainthatstuff.com/introduction-to-neural-networks.html>
5. Ben Krose, Patrick van der Smagt. An Introduction to Neural Networks. Amsterdam, November, 1996. 135 P.
6. David Kriesel. A Brief Introduction to Neural Networks.
http://www.dkriesel.com/en/science/neural_networks
7. David Stutz. Understanding Convolutional Neural Networks // Seminar Report, Fakultät für Mathematik, Informatik und Naturwissenschaften. August, 2014.
8. Ganesh K. Venayagamoorthy, Teaching Neural Networks Concepts and Their Learning Techniques // Proceedings of the 2004 American Society for Engineering Education Midwest Section Conference.
9. Ing.H.Ney, B.Leibe. Matching Algorithms for Image Recognition // RWTH Aachen University, Januar, 2010.
10. Keiron O'Shea, Ryan Nash. An Introduction to Convolutional Neural Networks // arXiv:1511.08458v2
11. Leon Bottou. Stochastic Gradient Descent Tricks // Neural Networks: Tricks of the Trade. Volume 7700 2012 of the series Lecture Notes in Computer Science. P. 421-436.
12. Line Eikvil. Optical Character Recognition. December, 1993.

13. Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, Rob Fergus. Deconvolutional Networks // Computer Vision and Pattern Recognition, June 13-18, 2010.
14. Neural Networks.
https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction to neural networks
15. Samer Hijazi, Rishi Kumar, Chris Rowen. Using Convolutional Neural Networks for Image Recognition. <http://www.cadence.com/en/default.aspx>
16. Si Chen, Yufei Wang. Convolutional Neural Networks and Convex Optimization, 2014.
17. Son Lam Phung, Abdesselam Bouzerdoum. MATLAB Library for Convolutional Neural Networks. Technical report.
<http://www.uow.edu.au/~phung/>
18. Tanzila Saba, Amjad Rehman, Ghazali Sulong. Cursive Script Segmentation With Neural Confidence // ICIC International ISSN 1349-4198. P. 4955-4964.
19. Understanding Convolutions.
<http://colah.github.io/posts/2014-07-Understanding-Convolutions/>
20. Vivek Shrivastava, Navdeep Sharma. Artificial Neural Networks Based Optical Character Recognition // Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.5, October 2012.
21. Yangwei Wu, Haouhua Zhao, Liqing Zhang. Image Denoising with Rectified Linear Units // Neural Information Processing. 21st International Conference, ICONIP 2014 / Eds. Chu Kiong Loo, Keem Siah Yap, Kok Wai Wong, Andrew Teoh, Kaizhu Huang. Kuching, Malaysia, November 3-6, 2014. P. 142-149.
22. Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner. Gradient-Based Learning Applied to Document Recognition // Proc. of the IEEE, November 1998.