АЗЕРБАЙДЖАНСКАЯ РЕСПУБЛИКА

На правах рукописи

АРХИТЕКТУРЫ И МОДЕЛИ НЕЙРОННЫХ СЕТЕЙ, РЕАЛИЗУЮЩИЕ МЕТРИЧЕСКИЕ МЕТОДЫ РАСПОЗНАВАНИЯ

Специальность: 3338.01. - Системный анализ, управление и обработка информации

Отрасль науки: техника

Соискатель: **Гейдаров Полад Шахмалы оглы**

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора наук

Баку – 2021

Диссертационная работа выполнена в Институте Систем Управления НАН Азербайджана, в лаборатории № 1.2 – «Методы распознавания сигналов и системы технической диагностики».

Научный консультант: доктор технических наук, профессор
Нусратов Октай Гудрят оглы

Официальные оппоненты:

член-корр. НАНА,
доктор технических наук, профессор
Мамедова Масума Гусейн гызы
доктор технических наук, профессор
Ибрагимов Байрам Ганимат оглы
доктор технических наук, профессор
Алиев Эльчин Рашид оглы
доктор технических наук, профессор
Мамедов Джаваншир Фирудин оглы

Диссертационный совет ED 1.20 Высшей Аттестационной Комиссии при Президенте Азербайджанской Республики, действующий на базе Института Систем Управления НАН Азербайджана

Председатель диссертационного совета:
академик НАНА, д.т.н., профессор
_____ **Аббасов Али Мамед оглы**

Ученый секретарь диссертационного совета:
д.т.н., профессор
_____ **Мусаева Наиля Фуад гызы**

Председатель научного семинара:
д.т.н., доцент
_____ **Пашаев Фархад Гейдар оглы**

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Нейронные сети — это системы искусственного интеллекта, имитирующие возможности биологического мозга. В настоящее время нейронные сети широко используются в различных задачах, но наиболее широкое распространение применение нейронных сетей получила в задачах распознавания образов. В этих задачах нейронные сети позволяют на основе обучающих входных и выходных выборок X, Y определять соответствие входного элемента к своему образу.

Нейронные сети имеют целый ряд преимуществ, по сравнения с другими методами распознавания. Нейронная сеть позволяет классифицировать объекты без необходимости создания и использования строгих алгоритмов распознавания для той или иной задачи. В настоящее время нейронные сети позволяют достичь наилучших результатов распознавания по сравнению с существующими другими алгоритмами распознавания. При технической реализации нейронные сети позволяют также достичь высокой скорости распознавания объектов, благодаря высокому уровню паралеллизации задачи распознавания выполняемых посредством нейронов. Вместе с тем существующие ныне классические архитектуры нейронных сетей имеют и целый ряд недостатков.

Существуют разные архитектуры нейронных сетей. В практических задачах наиболее часто используют многослойный персептрон. Особенностью многослойного персептрона является наличие более чем одного обучаемого слоя (как правило — два или три). Процедура создания этих сетей выявляет недостатки этих нейронных сетей, которые выглядят следующим образом:

1. При решении той или иной задачи сеть выбирается, исходя из постановки задачи и имеющихся данных для обучения. Для обучения с учителем требуется наличие для каждого элемента выборки «экспертной» оценки. Иногда

получение такой оценки для большого массива данных является просто невозможным.

2. После выбора общей структуры нужно экспериментально подобрать параметры сети. Для сетей, подобных персептрону, это будет число слоев, наличие или отсутствие обходных соединений, передаточные функции нейронов. При выборе количества слоев и нейронов в них следует исходить из того, что способности сети к обобщению тем выше, чем больше суммарное число связей между нейронами. С другой стороны, число связей ограничено сверху количеством записей в обучающих данных. Таким образом, в настоящее время для персептрона не существует точных алгоритмов, четко определяющих структуру и топологию нейронной сети.

3. Если в процессе обучения нейронной сети ошибка на обучающих данных продолжает уменьшаться, а ошибка на тестовых данных увеличивается, значит, сеть перестала выполнять обобщение и просто «запоминает» обучающие данные. Это явление называется переобучением сети. В таких случаях обучение обычно прекращают. В процессе обучения могут проявиться другие проблемы, такие как паралич или попадание сети в локальный минимум поверхности ошибок. Таким образом для классических сетей невозможно заранее предсказать проявление той или иной проблемы, равно как и дать однозначные рекомендации к их разрешению.

4. Нет также четкого понимания того, как те или иные настройки весовых и пороговых значений нейронов нейронной сети влияют на результат работы нейронной сети. В целом внутреннее содержание нейронной сети и в частности сети персептрона представляется как некий черный ящик, не поддающийся ясному пониманию.

5. Даже в случае успешного, на первый взгляд, обучения сеть не всегда обучается именно тому, чего от неё хотел создатель. Тестирование качества обучения нейросети необходимо проводить на примерах, которые не участвовали в её обучении. При этом число тестовых примеров должно быть

тем больше, чем выше качество обучения. Если ошибки нейронной сети имеют вероятность близкую к одной миллиардной, то и для подтверждения этой вероятности нужен миллиард тестовых примеров. Получается, что тестирование хорошо обученных нейронных сетей становится очень трудной задачей.

Многослойный персептрон является сетью прямого распространения, то есть сетью, который не имеет обратных связей. Архитектуры нейронных сетей, которые имеют обратные связи называются рекуррентными нейронными сетями. При этом под обратной связью подразумевается связь от логически более удалённого элемента к менее удалённому. Наличие обратных связей позволяет запоминать и воспроизводить целые последовательности реакций на один стимул. Такие особенности потенциально предоставляют множество возможностей для моделирования биологических нейронных сетей. Но этот класс сетей также имеет свои недостатки:

1. Большинство возможностей для этих сетей на данный момент плохо изучены в связи с возможностью построения разнообразных архитектур и сложностью их анализа.

2. Основным недостатком этих сетей является отсутствие устойчивости, а в случаях, когда она достигается, сеть становится эквивалентной однослойной нейронной сети из-за чего она не в состоянии решать линейно неразделимые задачи. В итоге ёмкость таких сетей крайне мала.

Приведем еще одну архитектуру нейронной сети, получившее популярность в последние годы - это архитектура нейронных сетей с глубинным обучением, а именно свёрточная нейронная сеть. Архитектура свёрточной нейронной сети заключается в чередовании свёрточных слоев и субдискретизирующих слоев.

Особенность данных сетей заключается в том, что процесс распознавания идет от слоя к слою от простых элементов-деталей изображения к сложным изображениям, понятиям и т.д. Преимущество архитектуры сверточных сетей заключается в

том, что сеть способна выделять и распознавать объекты на изображении и кроме того, в отличии от сети классического персептрона, в сверточных нейронных сетях не используется полносвязная архитектура сети. Связи нейронов определяются выборочно, что значительно уменьшает количество настраиваемых весовых значений нейронной сети. Несмотря на все преимущества и популярность которую получили эти сети, тем не менее разработка сверточных сетей имеют ряд сложностей и не определенностей, и в частности следующие:

1. Архитектура сверточных сетей весьма сложна и громоздка.

2. Слишком много варьируемых параметров сети, непонятно, для какой задачи и вычислительной мощности какие нужны настройки. Так, к варьируемым параметрам можно отнести: количество слоев, размерность ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, шаг сдвига ядра при обработке слоя, необходимость слоев субдискретизации, степень уменьшения ими размерности, функция по уменьшению размерности (выбор максимума, среднего и т. п.), передаточная функция нейронов, наличие и параметры выходной полносвязной нейросети на выходе свёрточной. Все эти параметры существенно влияют на результат, но выбираются исследователями эмпирически.

Еще одна архитектура нейронных сетей это импульсная нейронная сеть (ИмНС) или спайковая нейронная сеть. Это третье поколение искусственных нейронных сетей (ИНС), которое отличается тем, что в нем нейроны обмениваются короткими импульсами одинаковой амплитуды (у биологических нейронов — около 100 мВ). В настоящее время является самой реалистичной, с точки зрения физиологии, моделью ИНС. Особенностью данных сетей является то, что сигналы меду нейронами в нейронной сети распространяются в виде импульсов. Сеть получает на входы серию импульсов и выдаёт импульсы на выходе. В каждое мгновение каждый нейрон имеет некоторое значение (аналого-электрического потенциала у биологических нейронов) и, если это значение

превышает пороговое, то нейрон посылает одиночный импульс, после чего его собственное значение падает до уровня ниже среднего значения (аналог процесса реабилитации у биологических нейронов, так называемый рефрактерный период) на 2-30 мс. При выведении из состояния равновесия потенциал нейрона начинает плавно стремиться к среднему значению.

В частотных ИНС используется сигнал, который принимает значение, зависящее от частоты порождения импульсов определенной группой нейронов (веса нейронов, собственно, и являются формой представления этой частоты). Тем не менее, средняя частота импульсов в последовательности является довольно плохим вариантом представления информации, так как различные виды стимуляции могут приводить к одинаковой средней частоте импульсов.

Для избавления от этого недостатка в импульсных ИНС используются следующие виды представления информации.

1. фазовый (временной) — информация о сигнале задается точным (или в пределах некоторого окна) положением импульсов во времени (относительно какого-либо общего опорного ритма головного мозга);
2. синхронный позиционный - информация о сигнале задается синхронной активностью различных групп нейронов, и, как следствие, синхронным (или в пределах некоторого окна) появлением импульсов на определенных выходах сети (например, реагирующие на высокие и низкие частоты слуховые рецепторы улитки уха находятся в разных зонах);
3. время до появления первого импульса — информация о сигнале задается временем появления первого импульса на каком-либо выходе;
4. порядковый — информация о сигнале задается порядком получения импульсов на выходах сети;
5. интервальный — информация о сигнале задается расстоянием между импульсами, получаемыми на выходах сети;
6. резонансный — информация о сигнале задается плотной последовательностью импульсов (очередью), приводящей к

возникновению резонанса (одиночные импульсы затухают и не вносят никакого вклада в передачу информации).

Главный недостаток импульсных нейронных сетей (ИмНС) это отсутствие совершенных алгоритмов обучения, реализованных для этих сетей. Связано это с тем, что известные алгоритмы обучения (такие как back propagation) требуют необходимость дифференцирования сигналов связей, но дифференцирование импульсных наборов сигналов - является весьма сложной и до конца не понятной задачей.

Сложности, связанные с импульсными нейронными сетями, также связаны еще и с тем, что не достаточно понятно, как происходит кодирование и раскодирование импульсов в биологических системах, откуда собственно и было заимствована идея импульсной передачи сигналов в нейронных сетях. Придуманные способы кодирования, приведенные выше, не являются универсальными и имеют свои недостатки.

Еще один не достаток ИмНС это сложность реализации импульсных нейронных сетей, как в программной, так и в аппаратной реализации. В программной реализации весьма не практично реализация передача сигналов между нейронами в виде пучка импульсов. Это приводит к слишком сильному замедлению работы нейронной сети. В аппаратной среде реализация импульсных нейронных сетей требует создание и изобретение новых типов нейронов, что требует создание для одного нейрона сложных электронно-аппаратных систем.

Отдельно также нужно сказать про алгоритмы обучения нейронных сетей, и в частности метода обратного распространения ошибки. В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы продвинемся по нему, то ошибка уменьшится. Последовательность таких шагов (замедляющаяся по мере приближения к дну) в конце концов приведет к минимуму того или иного типа.

Алгоритмы обучения нейронных сетей также имеют свои проблемные стороны, которые на практике усложняют работу разработчиков нейронных сетей.

Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов. При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или (если поверхность ошибок имеет особо вычурную форму) уйти в неправильном направлении. Классическим примером такого явления при обучении нейронной сети является ситуация, когда алгоритм очень медленно продвигается по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при маленьком шаге, вероятно, будет схвачено верное направление, однако при этом потребуется очень много итераций и процесс обучения будет идти слишком медленно.

Несмотря на многочисленные успешные применения обратного распространения, оно не является универсальным решением. Больше всего неприятностей приносит неопределённо долгий процесс обучения. В сложных задачах для обучения сети могут потребоваться дни или даже недели, она может и вообще не обучиться. Причиной может быть одна из описанных ниже. В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях. Это явление называется параличом сети. Обычно этого избегают уменьшением размера шага, но это увеличивает время обучения. Различные эвристики использовались для предохранения от паралича или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные.

Следует также отметить проблему переобучения сети, что является скорее результатом ошибочного проектирования её топологии или неправильным выбором критерия остановки обучения. При переобучении теряется свойство сети обобщать информацию. Весь набор образов, предоставленных к

обучению, будет выучен сетью, но любые другие образы, даже очень похожие, могут быть распознаны неверно. Таким образом мы минимизируем не ту ошибку, которую на самом деле нужно минимизировать, - ошибку, которую можно ожидать от сети, когда ей будут подаваться совершенно новые наблюдения.

Описанные проблемы с локальными минимумами и выбором размера сети приводят к тому, что при практической работе с нейронными сетями, как правило, приходится экспериментировать с большим числом различных сетей, порой обучая каждую из них по нескольку раз (чтобы не быть введенным в заблуждение локальными минимумами) и сравнивая полученные результаты.

Перечисленные выше недостатки нейронных сетей и алгоритмов обучения можно обобщить и добавить еще и следующие недостатки:

1. Сложность выбора структуры архитектуры нейронной сети и ее параметров (количество нейронов, связей, слоев). Эти параметры нейронной сети выбираются либо эмпирически, либо при помощи приближенных выражений, либо имеют весьма сложную многослойную форму, как например, для сетей с глубоким обучением.

2. Сложность обучения нейронной сети, с большим количеством распознаваемых образов. При этом, чем больше количество распознаваемых образов, тем сложнее обучаются нейронные сети, и тем хуже, как правило, результат обучения. Это приводит к ограничению возможного количества образов (N), используемых в задачах распознавания с применением нейронных сетей.

3. В обучаемых моделях значения весов определяются на основе алгоритмов обучения, реализация которых требует длительного выполнения и очень большого объема обучающей выборки. При этом результаты обучения непостоянны и зависят не только от набора обучающей выборки и условий алгоритмов обучения, но и от параметров структуры сети.

4. Еще одна сложность — это проблемы с расширением уже обученной нейронной сети и добавления к ней новых

образов. Добавление новых образов к уже обученной сети требует выполнения процесса переобучения сети с полным или частичным изменением предыдущих весовых и пороговых значений, что также значительно усложняет введения изменений в уже обученную нейронную сеть.

В связи со всем выше сказанном, создание новых архитектур нейронных сетей, с новыми возможностями на сегодняшний день является весьма актуальной задачей. В связи с этим в данной работе предлагаются новые архитектуры нейронных сетей, которые могли бы позволить разрешить указанные проблемы типовых архитектур нейронных сетей.

Объект и предмет исследования. В данной работе объектом исследования являются сети прямого распространения (многослойный персептрон). В работе предлагаются новые архитектуры нейронных сетей прямого распространения, реализующие метрические методы распознавания, позволяющие значительно упростить и ускорить процедуру создания и обучения нейронных сетей прямого распространения. Возможности и способности этих сетей исследуются.

Основные цели и задачи работы. Основная цель работы заключается в создании технологий, алгоритмов и сетевых архитектур, позволяющих упростить и ускорить процедуру создания и обучения искусственных нейронных сетей, а также позволяющих сделать сети многослойного персептрона прозрачными и понятными. С этой целью в диссертационной работе были поставлены и решены следующие цели и задачи, заключающиеся в следующих пунктах:

1. Создание новых архитектур нейронных сетей на основе метрических методов распознавания, которые позволяют аналитически определять параметры сети: количество нейронов, слоев и связей на основе начальных условий задачи распознавания, таких как количество эталонов и образов.
2. Аналитическое определение и вычисление весовых и пороговых значений нейронной сети прямого распространения (многослойного персептрона).

Достижение этой цели выполняется путем реализации нейронной сети на основе существующих алгоритмов распознавания метрических методов распознавания.

3. Исследование полученных архитектур нейронных сетей, их возможностей и преимуществ.
4. Исследование возможностей применения предлагаемых нейронных сетей к различным типам задач, а также применения данных архитектур в многозадачных применениях.
5. Исследование возможностей использования предлагаемых архитектур нейронных сетей для задач нечеткой логики.
6. Определение диапазонов весовых значений, общих законов, условий и правил, определяющих распределение их значений.
7. Математическое обоснование связи предлагаемых нейронных сетей с архитектурой полносвязного многослойного персептрона, а также связи вычисленных весовых значений с весовыми значениями обученной нейронной сети при помощи классических алгоритмов обучения.
8. Исследование возможностей и способов представления различного рода входных объектов (изображений, символов, текстов, фотографий) для представления их в предлагаемую нейронную сеть.
9. Создание программного модуля, позволяющего: реализовывать предлагаемые архитектуры нейронных сетей, аналитически вычислять значения весов и другие параметры нейронной сети, выполнять обучение нейронной сети классическими алгоритмами обучения (back propaqation) и просматривать все значения и параметры нейронной сети.
10. Выполнение различных экспериментов с предлагаемыми нейронными сетями, в том числе выполнения сравнительных экспериментов возможностей предлагаемых архитектур нейронных сетей с

возможностями известных классических нейронных сетей, такими как полно связный многослойный персептрон, на базе цифр MNIST (60000 обучающей выборки и 10000 контрольной выборки), с целью анализа возможностей и преимуществ предлагаемых технологий с известными классическими нейросетевыми технологиями.

11. На основе предложенных технологий создание технологии определения значений весов нейронной сети без вычисления и обучения значений весов. Здесь ставится цель реализовать и описать такую возможность при помощи физических выражений электростатического поля и экспериментально подтвердить такую возможность.

Применяемые методы. В работе используются метрические методы распознавания, теория нейросетевых технологий, теория нечеткой логики, алгоритмы обучения нейронных сетей, методы обработки изображений, математическое моделирование и программирование, теория основ электростатики.

Основные положения, выносимые на защиту.

- решение, позволяющее сразу аналитически вычислять и определять значения весов и порогов нейронной сети многослойного персептрона;
- решение, позволяющее сразу аналитически определить структуру нейронной сети многослойного персептрона, на основе начальных условий задачи распознавания;
- оценка времени создания нейронной сети и аналитического вычисления значений всех весов;
- решение, позволяющее получить прозрачную нейронную сеть многослойного персептрона;
- решение, позволяющее уменьшить вероятность попадания нейронной сети в локальные минимумы;
- решение, позволяющие уменьшить объем обучающей базы для обучения многослойного персептрона;
- решение, позволяющее оптимизировать структуру нейронной сети, минимизировав количество нейронов и связей

нейронной сети;

- решения, позволяющие добавлять новые образы и задачи в нейронную сеть без изменения предыдущих весовых и пороговых значений.
- выполненные теоритические и практические доказательства, показывающие ускорение процедуры создания и обучения нейронной сети многослойного персептрона на основе архитектур предлагаемых архитектур нейронных сетей прямого распространения;
- оценка времени обучения и полученной результативности предлагаемой нейронной сети с классическими схемами обучения нейронных сетей многослойного персептрона.
- решение, позволяющее практически мгновенно определять значения весов многослойного персептрона с использованием параметров электростатического поля;
- созданный программный модуль в программной среде Builder C++ и реализующий предлагаемые архитектуры нейронных сетей, а также их возможности и позволяющий также сравнивать возможности полученных архитектур с возможностями классических нейронных сетей;

Научная новизна. Научная новизна предлагаемой работы заключается в следующих положениях:

1. Предлагается целый класс новых архитектур нейронных сетей.

2. Показана прямая связь между классическими архитектурами нейронных сетей и метрическими методами распознавания.

3. Показаны точные аналитические выражения, определяющие структуру нейронной сети, исходя из начальных условий задачи.

4. Показаны аналитические выражения, вычисляющие весовые и пороговые значения нейронов на основе выражений мер близости.

5. Показана, что предлагаемая технология создания нейронных сетей позволяет быстро применять нейронные сети

прямого распространения в многозадачных применениях.

6. Показана связь полученных сетей с классическим многослойным персептроном, а также возможность до обучения полученных сетей классическими алгоритмами обучения.

7. Показано, что для предлагаемых архитектур нейронных сетей процесс создания и обучения нейронной сети происходит на много быстрее по сравнения с классическими схемами обучения.

8. Показаны возможности расширения полученных сетей с добавлением новых образов в задачу распознавания, без необходимости изменения предыдущих параметров.

9. Предлагаемая технология создания вычисляемых нейронных сетей прямого распространения позволяет обучать нейронные сети с меньшим количеством обучающей выборки.

10. Предлагаемая технология вычисляемых нейронных сетей прямого распространения позволяет значительно уменьшить в процессе обучения вероятность паралича сети или попадания нейронной сети в локальный минимум.

11. Показано, что значение весов предлагаемой архитектуры нейронной сети могут быть определены практически мгновенно (без вычислений и обучений) при помощи параметров электростатического поля.

Практическая ценность работы.

1. Предложенные алгоритмы и архитектуры нейронных сетей позволяют ускорить создание и обучение классических нейронных сетей. Это обеспечивается благодаря возможности аналитического вычисления количества нейронов, слоев и связей нейронной сети, а также вычисление значений весов и порогов нейронов и связей нейронной сети. Эти возможности позволяет быстро создавать работоспособную нейронную сеть. Проведенные эксперименты показали, что процесс вычисления нейронной сети выполняется за доли секунд. Процесс до обучения предлагаемых нейронных сетей позволяет ускорить процесс до обучения на 40 и более процентов быстрее, чем

выполняется обучение классических нейронных сетей.

2. Предлагаемые нейронные сети также позволяют легко расширять нейронную сеть с добавлением в задачу распознавания новых распознаваемых образов без необходимости изменения предыдущих весовых и пороговых значений, что особо будет актуально для задач распознавания с использованием большого количества распознаваемых образов.

3. Предлагаемые нейронные сети – прозрачны, что также позволяет оптимизировать структуру нейронной сети без изменения результативности нейронной сети.

4. Предлагаемые технологии вычисляемых нейронных сетей позволяет также значительно уменьшить вероятность паралича сети и попадания сети в локальный минимум в процессе обучения.

5. Предлагаемые нейронные сети позволяют уменьшить количество требуемого объема обучающей выборки.

6. Предлагаемые нейронные сети позволяют упростить и ускорить процедуру использования одной нейронной сети в многозадачных применениях.

7. Предлагаемые нейронные сети прозрачны, что может позволить понять алгоритм распознавания в обученной нейронной сети прямого распространения.

Апробация работы. Основные результаты диссертационной работы прошли апробацию на Республиканских и Международных конференциях, таких как: Труды VII Международного симпозиума «Интеллектуальные системы», INTELS-2006, Краснодар, Россия; Труды Международной Конференции PCI – 2006 «Проблемы Кибернетики и информатики», Баку, 2006; The second international conference “Problem of Cybernetics and informaties.”, Bakı, 2008; Elektron elm problemləri üzrə 1 respublika elmi-praktiki konfransın materialları, Bakı, 2012; Международная научно-практическая конференция “Научно-практические исследования”, Россия, Омск, 2020; XXVI Международная

научно-практическая конференция «Advances in Science and Technology», Россия, Москва, 2020.

Применение работы. Предлагаемая технология была применена и многократно проверена для задачи распознавания рукописных арабских цифр на базе MNIST (Modified National Institute of Standards and Technology), состоящей из 60000 изображений цифр для обучающей выборки и 10000 изображений цифр для контрольной выборки. Применение этой работы было выполнено в программной среде Pyton и Builder C++.

Изданные труды. По теме диссертационной работы было издано 30 работ, из них 23 статьи. 26 работ изданы были без соавторов, 21 трудов были изданы в зарубежных рецензируемых научных журналах, из которых 19 в журналах рекомендуемых ВАК-ом России, 12 входят в базу Scopus, 15 входят в базу Web of Science и 5 статей опубликованы редакцией Springer. Перечень изданных трудов приведен в конце автореферата.

Наименование учреждения, где выполнена диссертационная работа. Диссертационная работа выполнена в Институте Систем Управления Национальной Академии Наук Азербайджана.

Структура и объем диссертации. Диссертационная работа состоит из Введения (32 стр. - 52415 знака), 8 глав (1 глава - 54 стр. - 50834 знака, 2 глава – 14 стр. – 13922 знака, 3 глава – 24 стр. – 26234 знака, 4 глава – 41 стр. – 42945 знака, 5 глава – 54 стр. – 53398 знака, 6 глава – 13 стр. – 14401 знака, 7 глава – 23 стр. – 24870 знака, 8 глава – 30 стр. – 38516 знака), Итоговых выводов (3 стр. – 4798 знака) и Приложения. В диссертационной работе приведены 161 рисунков и 24 таблицы. Основной объем диссертации составляет 299 страниц (322480 знака) машинописного текста без учета пробелов в тексте.

СОДЕРЖАНИЕ ДИССЕРТАЦИОННОЙ РАБОТЫ

В ведении раскрывается актуальность работы. Рассматриваются наиболее значимые и используемые классические архитектуры нейронной сети и классические алгоритмы обучения, а также слабые и сильные стороны этих архитектур. На основе существующих проблемных сторон нейронных сетей формируется цель работы.

В главе I, состоящей из 21-го параграфа, приводится общее пояснение метрических методов распознавания и описание некоторых алгоритмов, реализующих метрические методы распознавания, а также предлагаются архитектуры нейронных сетей, реализующие метрические методы распознавания [8, 15, 25]. Приводятся аналитические выражения, определяющие весовые значения первого слоя, а также аналитические выражения, определяющие структуру нейронной сети: количество нейронов, слоев, связей [8, 15]. Предлагается также алгоритм отбора минимального набора эталонов для нейронной сети [8, 14]. Также приводится общий алгоритм аналитического вычисления всех значений весов в таблице весов [26, 27]. Предлагаются способы ускорения данного алгоритма. Выполняется оценка времени вычисления всех значений весов нейронной сети.

В параграфе 1.1. говорится о гипотезе компактности, на основе существования которого базируются метрические методы распознавания.

В параграфе 1.2. приводится описание алгоритма метода построения эталонов (метод эталонов).

В параграфе 1.3. приводится описание алгоритма "метода ближайшего соседа".

В параграфе 1.4. приводится описание алгоритма "метода потенциальных функций".

В параграфе 1.5. приводятся примеры аналитических выражений, определяющих весовые значения нейронов первого слоя, [15, 21, 26] например, в простейшем случае это может быть выражение:

$$w_i = \left(y_{etal1\,A} - y_{testi}\right)^2 - \left(y_{etal2\,B} - y_{testi}\right)^2, \qquad (1)$$

где $y_{etal1\,A}$ и $y_{etal2\,B}$ значения кривых *etal*$_1$ и *etal*$_2$ для столбца *AB* на рис.1.

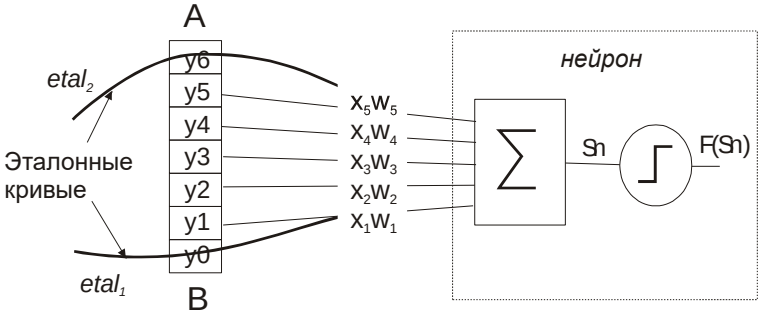


Рис. 1 Схема получения одного нейрона

Например, если значение $y_{etal1\,A}$ = 6, $y_{etal2\,B}$ = 0 (Рис. 1), то для разных значений y_{testi} получим веса w_i для всех участков между двумя кривыми в столбце *AB*:

$$\begin{gathered} w_1 = (6\text{-}1)^2 - (0\text{-}1)^2 = 25 - 1 = 24, \\ w_2 = (6\text{-}2)^2 - (0\text{-}2)^2 = 16 - 4 = 12, \\ w_3 = (6\text{-}3)^2 - (0\text{-}3)^2 = 9 - 9 = 0, \\ w_4 = (6\text{-}4)^2 - (0\text{-}4)^2 = 4 - 16 = -12, \\ w_5 = (6\text{-}5)^2 - (0\text{-}5)^2 = 1 - 25 = -24, \end{gathered} \qquad (2)$$

В параграфе приводится также выражение, определяющее значение состояния нейрона первого слоя,(Рис.1):

$$Sn = \sum_{j=0}^{M}\sum_{i=0}^{K} x_{ij} w_{ij}, \qquad (3)$$

где x_{ij} значение *x* для *i*-той строки и *j*-того столбца, w_{ij} - вес для *i*-той строки и *j*-того столбца; *M*, *K* – количество столбцов и строк изображения (Рис. 1).

В параграфе 1.6. приводится архитектура двух слойной нейронной сети (Рис. 2), реализующей "метод построения эталонов" (метод эталонов) [8], а также рассматриваются функции активации нейрона первого (4) и второго слоя (5):

$$f(Sn) = 1, \text{ если } Sn > 0; \quad (4)$$
$$f(Sn) = 0, \text{ если } Sn < 0;$$

где $f(Sn)$ – функция активации нейрона первого слоя (Рис. 1, 2) и

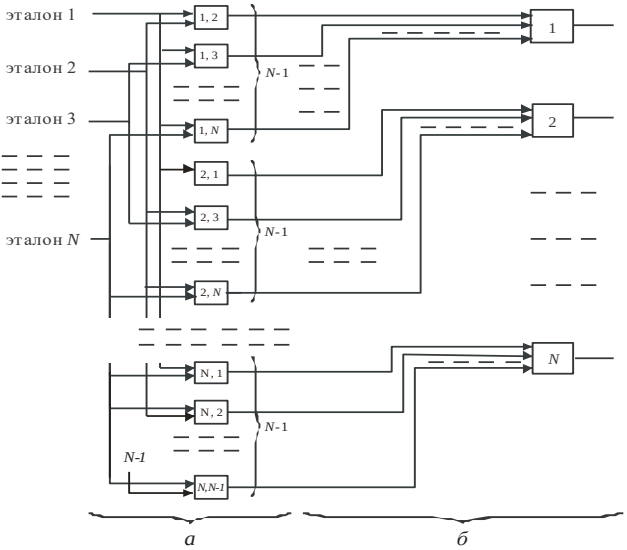


Рис. 2 Двухслойная нейронная сеть с не минимизированной архитектурой.

$$f(Sn_i) = 1, \text{ если} Sn_i >= \alpha(N-1), \quad (5)$$
$$f(Sn_i) = 0, \text{ если} Sn_i < \alpha(N-1),$$

где Sn_i – значение i-го нейрона второго слоя, определяемого по выражению:

$$Sn_k^{(2)} = \sum_{j=1, j\neq k}^{N-1} \left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)} \right) \right), \quad (6)$$

α- значение веса w_i. В наиболее простом случае $\alpha = 1$. N – количество эталонных образцов. При этом количество нейронов первого слоя (n) нейронной сети на рис.2 определяется по выражению:

$$n = N(N-1) \quad (7)$$

а количество нейронов второго слоя равно количеству эталонов N.

В параграфе 1.7. приводится архитектура двухслойной нейронной сети в сжатой форме рис.3 [8]. Это схема нейронной сети позволяет уменьшить количество нейронов (n) первого слоя в 2 раза (9), исходя из того, что из нейронной сети исключаются нейроны, выполняющие разделение одинаковых пар эталонов, например {«эталон1», «эталон2»} и {«эталон2», «эталон1»}. При этом во втором слое выходы исключенных нейронов первого слоя заменяются на инвертированные значения предыдущих нейронов (Рис. 3b). Функция состояния нейрона первого слоя определяется по выражению [15]:

$$Sn^{(2)}{}_k = \alpha\left(\sum_{i=1}^{k-1} \overline{f}\left(Sn^{(1)}{}_{i_k}\right) + \sum_{i=k+1}^{N} f\left(Sn^{(1)}{}_{k_i}\right)\right) \qquad (8)$$

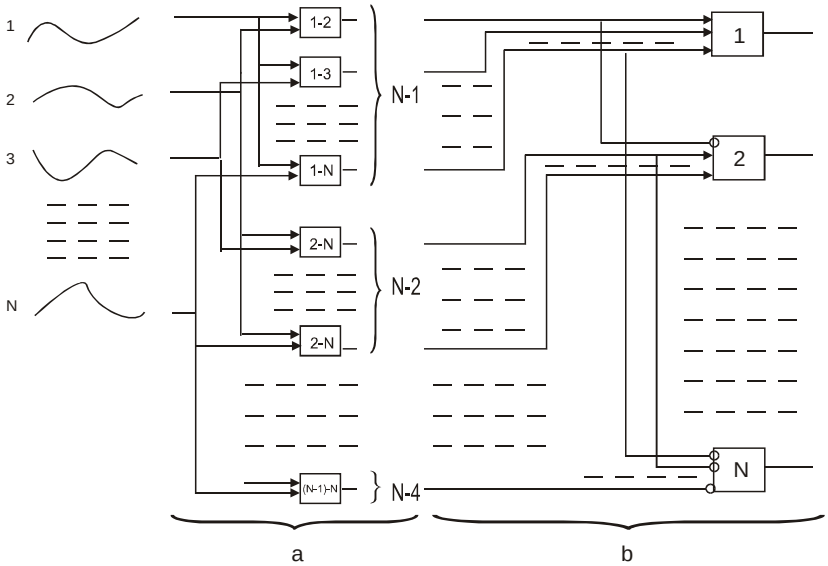


Рис. 3 Схема двухслойной нейронной сети в сжатой форме для распознавания *N* кривых.

$$n = C_N^2 = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2} \qquad (9)$$

В параграфе 1.8. рассматривается трехслойная архитектура, реализующая метод ближайшего соседа. Для этого к схемам на

рис. 3 добавляется третий слой (Рис. 4с) каждый нейрон которого объединяет эталоны одного образа [21].

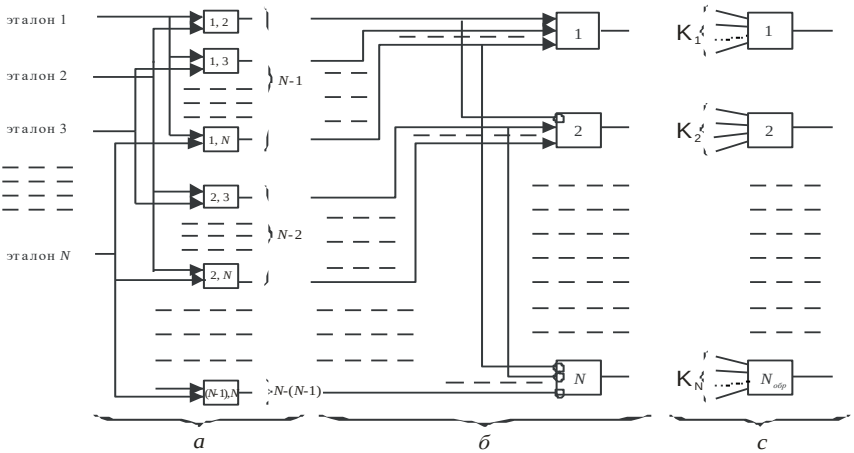


Рис. 4 Трехслойная схема с архитектурой нейронной сети на основе метрических методов распознавания.

Функция активации нейрона третьего слоя определится по выражению:

$$f^{(3)}(Sn^{(3)}_k) = 1, \text{ если } Sn^{(3)}_k > 0; \quad (10)$$
$$f^{(3)}(Sn^{(3)}_k) = 0, \text{ если } Sn^{(3)}_k = 0;$$

где $Sn^{(3)}_k$- значение k–го нейрона третьего слоя, вычисляющего по выражению

$$Sn_k^{(3)} = \sum_{j=1}^{K} \left(\alpha \times f\left(Sn_j^{(2)} \right) \right), \quad (11)$$

где j - указывает на выходы нейронов второго слоя, принадлежащих k-му образу, K- количество эталонов, принадлежащих k-му образу.

В параграфе 1.9. рассматривается возможность дополнительного сжатия нейронной сети, путем исключения нейронов первого слоя [15], которые выполняют разделение эталонов одного образа. При этом показано, что количество нейронов первого слоя (n) уменьшится и определится следующим образом:

$$n = n_1(N_{etal} - n_1) + n_2(N_{etal} - n_1 - n_2) + \dots$$
$$+ n_j(N_{etal} - \sum_{i=1}^{j} n_i) + \dots + n_{N-1}(N_{etal} - \sum_{i=1}^{N-1} n_i)$$,

$$n = \sum_{j=1}^{N-1} n_j (N_{etal} - \sum_{i=1}^{j} n_i), \qquad (12)$$

где n_j, n_i - количество эталонов *j*-го и *i*-го образа, N_{etal} - общее количество всех эталонов, *N* – количество образов.

В параграфе 1.10. приводятся примеры представления эталонов, и примеры метрических выражений, которые могут быть использованы для вычисления таблиц весов нейронов первого слоя [8, 15, 21].

Например, для черно белых изображений символов возможно использовать геометрическую евклидовую меру близости. В этом случае для каждой точки поверхности изображения определяется расстояние до каждого графического образа. По значениям этих данных составляется таблица весов для каждой пары образов. На рис. 5 приведен пример для одной точки изображения (x_t, y_t) по отношению к графическим образам «7» и «2».

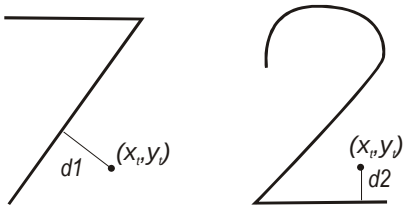


Рис. 5 Расстояния d_1 и d_2 для точки (x_t, y_t).

Здесь для точки (x_t, y_t) наименьшие расстояния до каждого образа будут равно d_1 и d_2. Соответственно вес w_{ij} для этой точки будет определяться как:

$$w_{ij} = d_1^2 - d_2^2 \ , \qquad (13)$$

где d_1^2 и d_2^2 - заменяет $\left(y_{etal1\,j} - y_{testij}\right)^2$, $\left(y_{etal2\,i} - y_{testij}\right)^2$ и определяются соответственно выражениями:

$$d_1^2 = \left(x_1 - x_t\right)^2 + \left(y_1 - y_t\right)^2, \quad (14)$$
$$d_2^2 = \left(x_2 - x_t\right)^2 + \left(y_2 - y_t\right)^2,$$

где (x_1, y_1) и (x_2, y_2) ближайшие точки эталонных образов до точки (x_t, y_t) (Рис. 5).

В параграфе 1.11. с целью уменьшения количества вычисляемых таблиц весов в нейронную сеть добавляется нулевой слой (Рис. 6). Здесь функция активации для нейрона нулевого слоя определится как:

$$f\left(Sn_{0,k}\right) = Sn_{0,k}, \quad (15)$$

где $Sn_{0,k}$ - значение k-го нейрона нулевого слоя, определяемого по выражению:

$$Sn_{0,k} = \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} w_{ij,k}, \quad (16)$$

где $w_{ij,k}$ - значение веса i-го столбца и j-ой строки k–ой таблицы весов, определяемой характеристикой меры близости применяемой в методе.

При этом таблицы весов определяются только для каждого нейрона нулевого слоя (Рис. 7). Поскольку количество нейронов нулевого слоя равно количеству эталонов, то соответственно количество вычисляемых таблиц весов также уменьшится со значений, определяемых в выражениях (7), (9), (12) до количества эталонов $N_{этал}$. Значение весов для таблиц нулевого слоя будет определяться исходя из одного эталона, например, в простейшем случае по выражению:

$$w_i = \left(y_{etal} - y_{\ i}\right)^2 \quad (17)$$

либо по выражению

$$w_{i,j}^{(0)} = d_1^2 = \left(x_i - x_t\right)^2 + \left(y_j - y_t\right)^2, \quad (18)$$

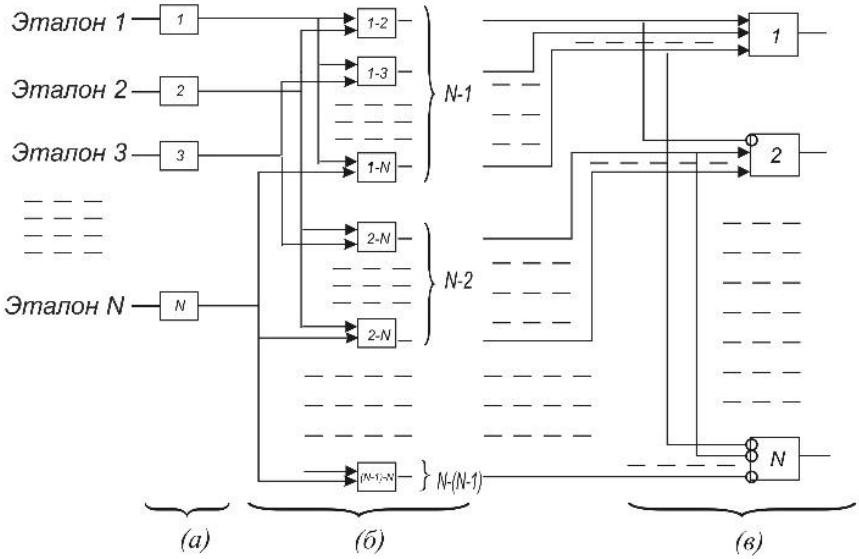


Рис. 6 Добавление нулевого слоя: *(а)* – нулевой слой, *(б)* – первый слой, *(в)* – второй слой.

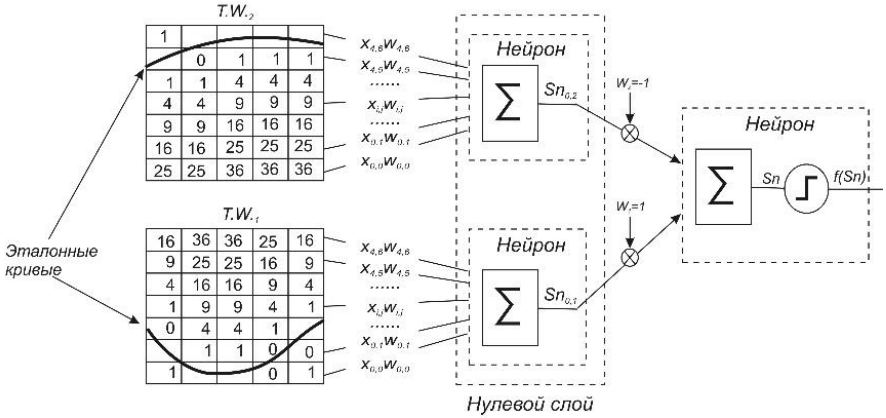


Рис. 7 Разделение нейрона первого слоя на два нейрона нулевого слоя.

В параграфе 1.12. предлагается алгоритм отбора минимального набора эталонов [15] для предлагаемых

нейронных сетей. Если не возможно явно выделить эталоны, то для отбора эталонов для данной задачи можно воспользоваться существующими алгоритмами обучающего отбора эталонов. В данном параграфе предлагается также алгоритм отбора эталонов, работа которого заключается в следующем. В качестве эталона в базу данных поочередно назначаются элементы тестируемой выборки (Рис. 8) для данного образа. После каждого назначения выполняется полное распознавание для всех остальных элементов выборки, результаты которого фиксируются в виде кодовой строки состоящей из нулей и единиц, где 1 соответствует идентифицированному элементу, 0 - не идентифицированному (табл. 1). Если количество единиц в строке соответствует необходимой достоверности распознавания, то в качестве эталона для этого образа выбирается текущий элемент выборки, на этом алгоритм завершается. Если же после аналогичной процедуры для всей выборки необходимый эталон не определяется, то выбирается кодовая строка с наибольшим количеством единиц в строке, которая помещается в первую строку обучающей таблицы (табл. 1), а элемент соответствующей этой строке помещается в базу эталонных данных. Далее в этой строке определяются номера столбцов, для которых значения равны нулю, что соответствует номерам элементам выборки, для которых распознавание было не удачно. В дальнейшем процедура повторяется аналогичным образом, но только теперь для этих не идентифицированных элементов. Каждый из этих элементов поочередно добавляется в базу эталонных данных, вычисляется кодовая строка, которая добавляется в обучающую таблицу (табл. 1). Итоговая полученная обучающая таблица становится объектом исследования на предмет определения наименьшего набора строк, поразрядное логическое сложение которых дает наибольшее количество единиц в итоговой кодовой строке (табл. 1), значение которого должно быть не меньше минимально требуемого значения. В результате работы данного алгоритма получаем минимальный набор эталонов, применение которых обеспечивает необходимый уровень достоверности для

этой выборки. Нужно отметить, что данный алгоритм может применяться для метода эталона вне зависимости от того, какой набор признаков или, какая характеристика меры близости используется в методе.

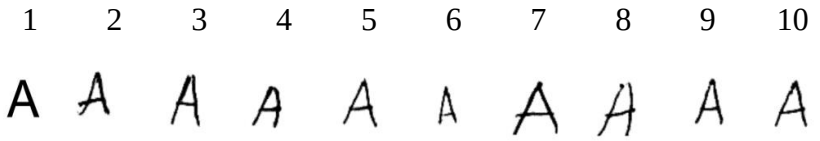


Рис. 8 Выборка из десяти (*n*=10) рукопечатных символов «А».

Табл. 1 Обучающая таблица для символа "*А*"

добавляемые в базу символы											
А – печ.	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	7
А-2	1	1	1	0	1	0	1	1	1	0	7
А-4	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**	7
А-8	1	0	1	1	1	0	0	1	1	1	7
А–печ., А-4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	10

В параграфах 1.13.-1.16 приводятся примеры демонстрирующие вычисления таблиц весов и на основе приведенных схем приводятся примеры создания нейронной сети, а также приводится пример реализации нейронной сети для базы рукопечатных символов и цифр.

На рис. 9 приведен пример вычисления значений фрагмента таблицу весов для двух эталонных кривых. В качестве выражения аналитического вычисления значений весов использовалась выражение (1).

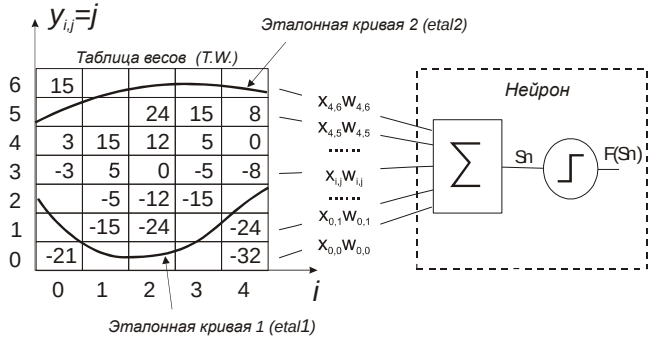


Рис. 9 Таблица весов для двух кривых.

Для определения значения весов таблицы весов для задачи распознавания графических символов может использоваться характеристика, определяющее наименьшее геометрическое расстояние до образа (13), (14), (Рис. 10*ab*).

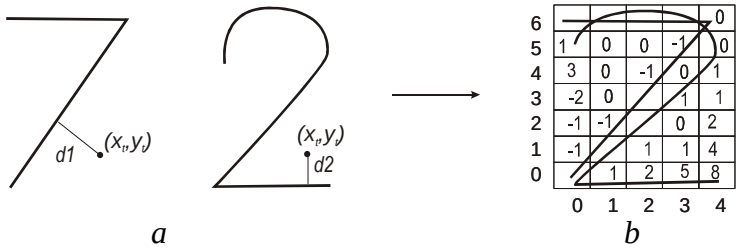


Рис. 10 Таблица весов для двух символов.

Пример создания простейшей нейронной сети приведен на рис.12. В данном примере создается двухслойная нейронная сеть для задачи распознавания 3-ех образов {7,2,4}. Таблицы весов размерности 5:7 для выделенных эталонов образа «7», «2» и «4» (Рис. 11), определены на основе выражений значений весов (13), (14) для нейрона нулевого слоя а значения весов определялись геометрической мерой близости (18).

В табл. 2 приведен результат работы сети в результате тестирования тестируемого символа «2» (Рис. 12), а также значения и выходы каждого нейрона.

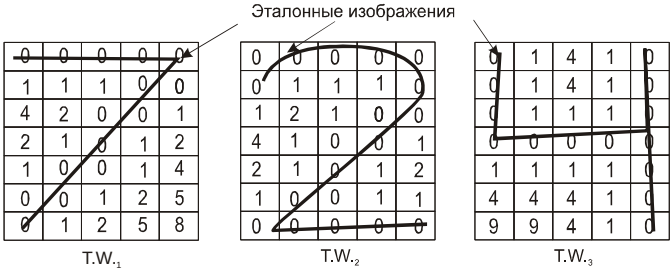


Рис. 11 Таблицы весов трех символов.

Табл. 2 Параметры нейронной сети на рис. 12.

Первый слой			Второй слой			*Yout*
k_m	Sn_{1,k_m}	$f(Sn_{1,k_m})$	*k*	$Sn_{2,k}$	$f(Sn_{2,k})$	
1-2	20	0	1	$0+1=1$	0	-
1-3	-8	1	2	$\overline{0}+1=2$	1	{2}
2-3	-28	1	3	$\overline{1}+\overline{1}=0$	0	-

По результатам таблицы можно видеть, что в итоге на выходах нейронной сети рис. 12 активизируется второй выход сети соответствующий символу образа {2}.

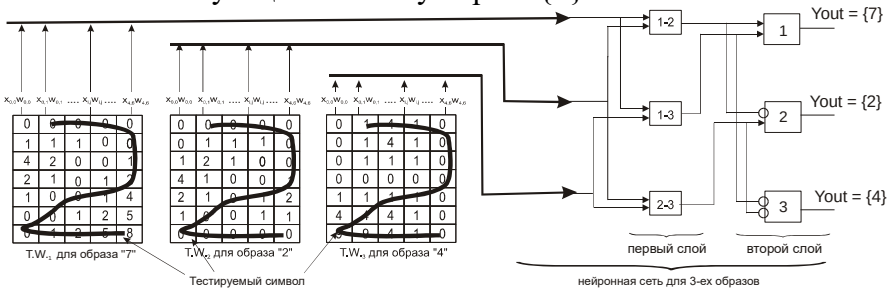


Рис. 12 Распознавание тестируемого символа «2» с использованием двухслойной сети для распознавания трех образов.

В параграфе 1.17. приводится описание алгоритма вычисления всех значений таблиц весов [26, 27], а также описание работы реализованного в среде Builder C++ на основе этого алгоритма программного модуля (Рис. 15).

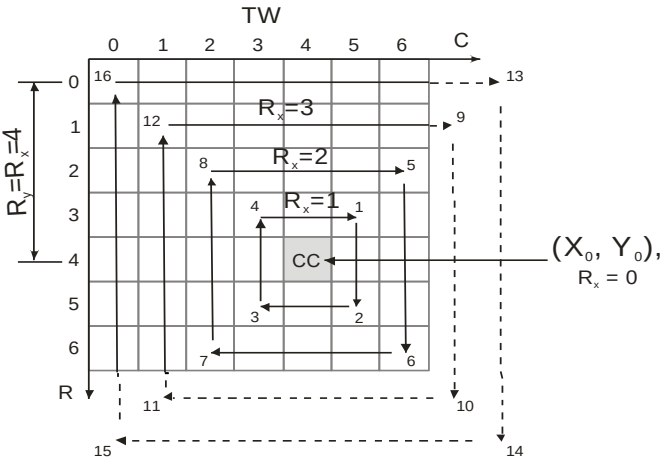


Рис. 13 Поиск по контурам активной ячейки относительно текущей ячейки (CC) в таблице весов (TW).

Процедура просмотра ячеек таблицы весов относительно текущей ячейки (CC) с координатами (4, 4)) выглядит следующим образом (Рис. 13). Сначала проверяется текущая ячейка. Далее выполняется просмотр ближайших ячеек таблицы весов располагаемых вокруг текущей ячейки. Движение происходит по замкнутому контуру, как показано на рис. 13. Контур движения в таблице представляет из себя квадрат. Сама процедура просмотра одного контура реализована с использованием четырех последовательных циклов, где каждый цикл выполняет просмотр ячеек по одной стороне квадрата: первый цикл - правая сторона квадрата, выполняет движение сверху вниз с точки 1 в точку 2 (Рис. 13); второй цикл - нижняя сторона квадрата, выполняется движение с права налево с точки 2 в точку 3; третий цикл - левая сторона квадрата, выполняется движение снизу вверх с точки 3 в точку 4; четвертый цикл - верхняя сторона квадрата, выполняет движение слева на право сточки 4 в точку 1. На рис. 13 *X0, Y0* – координаты текущей ячейки, $X_0 = 4$, $Y_0 = 4$. *Rx, Ry* –радиус контура по оси X и Y относительно текущей ячейки (Рис. 13). Начальные значения *Rx,*

Ry принимаются равными единице. После просмотра одного контура, выполняется увеличение значений *Rx* и *Ry* на единицу, после чего повторяется просмотр ячеек по замкнутому контуру, только теперь уже с новыми значениями *Rx=Ry*. Для каждой ячейки таблицы весов, встречаемой на пути движения вдоль замкнутого контура, выполняется проверка активности данной ячейки таблицы. Используя координаты выбранной ячейки в изображении *X1, Y1* сканируются все пиксели в данной ячейке на наличие затемненных участков, значение которых не выше 50, что соответствует затемненным оттенкам черно белого изображения, где тональность пикселя измеряется в диапазоне от 0 до 255. При выполнении условия затененности пикселей в ячейки таблицы, процесс прерывается и возвращается значение 1, что соответствует активности ячейки. Если после сканирования всей ячейки не определяется активные пиксели, то возвращается значение 0, что соответствует не активности данной ячейки. Таким образом, просматриваются все ячейки и для текущей ячейки и определяется ближайшая активная ячейка для данного эталона (примеры на рис. 14).

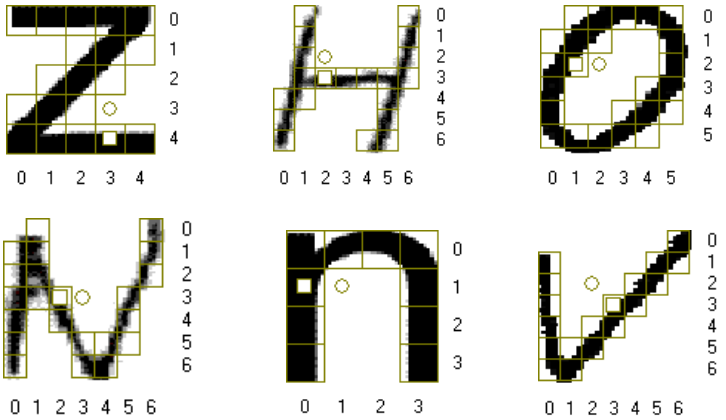


Рис. 14 Результаты определения ближайщей активной ячейки (отмечен белым квадратом) относительно текущей ячейки (отмечен белым кругом).

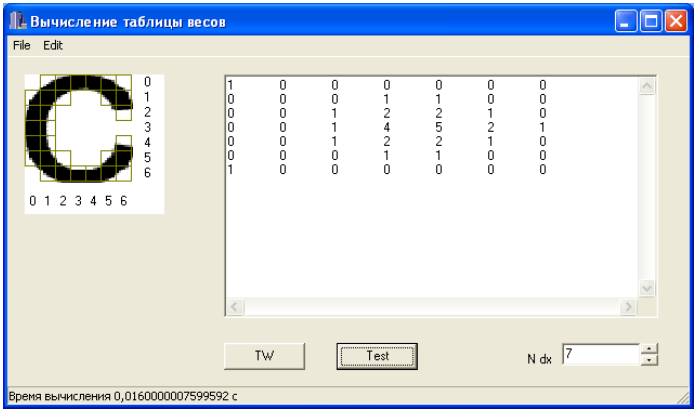


Рис. 15 Программный модуль вычисления таблицы весов нейрона нулевого слоя.

В параграфе 1.18. описан способ ускорения алгоритма вычисления всех значений таблиц весов [26], реализованного также в программном модуле.

В приведенном выше параграфе приводится алгоритм, в котором выполняется перебор всех ячеек таблицы выбранной размерности рис. 13. Как выше было показано для текущей ячейки таблицы выполняется поиск активных ячеек по контурам путем также перебора всех ячеек таблицы весов. При этом время вычисления таблиц весов может быть значительно меньше, в случае если в предложенном алгоритме исключить рассмотрение не нужных контуров и соответственно не нужных ячеек таблиц весов. В частности если сама текущая ячейка с координатами таблицы *(X0, Y0)* является активной, то нет необходимости просмотра всех остальных ячеек.

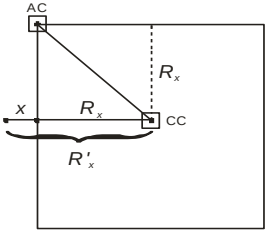


Рис. 16 Определение *x* – минимально достаточного количества просматриваемых контуров, после выявления активной ячейки *AC*.

Кроме того, если минимальное значение веса определяется на каком то контуре, то в этом случае нет также необходимости рассматривать все последующие контуры и соответственно ячейки таблицы в этих контурах. В этом случае необходимо определить количество *x* необходимых для просмотра последующих контуров (Рис. 16). Поскольку максимальное удаление активной ячейки (на рис. 16 *AC*) при текущем значении R_x от центра квадрата (текущей ячейки на рис. 16 *CC*) будет в вершинах квадрата и если при этом для вычисления значения веса используется метрическая мера квадрата евклидового расстояния, то для выявления значения *x* необходимо решение следующего неравенства:

$$(R_x + x)^2 < R_x^2 + R_x^2 \,, \tag{19}$$

$$x^2 + 2R_x x - R_x^2 < 0 \,. \tag{20}$$

Для решения неравенства определяем корни квадратного уравнения:

$$x^2 + 2R_x x - R_x^2 = 0. \tag{21}$$

Для этого определяем дискриминантное значение *D* и значения x_1, x_2:

$$x_1 = \left(-2R_x\sqrt{2} - 2R_x\right)/2 = -R_x\left(\sqrt{2} + 1\right), \tag{22}$$

$$x_2 = \left(2R_x\sqrt{2} - 2R_x\right)/2 = R_x\left(\sqrt{2} - 1\right). \tag{23}$$

Решением не равенства (19) и (20) является множество значений

$$-R_x\left(\sqrt{2} + 1\right) < x < R_x\left(\sqrt{2} - 1\right), \tag{24}$$

а максимальное необходимое значение

$$x_{max} = R_x\left(\sqrt{2} - 1\right). \tag{25}$$

Отсюда определяем необходимое максимальное значение R'_x (Рис. 16), которое может быть не больше $R_x\sqrt{2}$, что следует из выражения

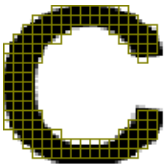
$$R'_x < x + R_x = R_x\left(\sqrt{2} - 1\right) + R_x = \\ = R_x\sqrt{2} \approx R_x \times 1{,}414. \tag{26}$$

Поскольку значение $\sqrt{2} < 1{,}5 = 3/2$, то для упрощения вычисления можно определить условие проверки как:

$$R'_x < 3R_x/2 \,. \tag{27}$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	29	20	13	8	4	1	0	0	0	0	1	4	8	13	20	29
1	26	17	10	5	2	1	0	0	0	0	1	2	5	10	17	25
2	25	16	9	4	1	0	0	0	0	0	0	1	4	8	13	20
3	20	13	8	4	1	0	0	0	0	0	0	1	2	5	10	17
4	17	10	5	2	1	0	0	0	1	0	0	0	1	4	9	16
5	13	8	4	1	0	0	0	1	2	1	0	0	1	4	8	13
6	10	5	2	1	0	0	1	2	4	1	0	0	1	2	5	10
7	9	4	1	0	0	0	1	4	4	1	0	0	0	1	4	8
8	8	4	1	0	0	0	1	4	4	1	0	0	0	1	2	5
9	5	2	1	0	0	0	1	1	1	1	1	0	0	0	1	4
10	4	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2
11	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	1	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1
13	1	0	0	0	1	4	4	4	4	4	4	1	0	0	0	1
14	1	0	0	1	2	5	9	9	9	9	5	2	1	0	0	0
15	0	0	0	1	4	8	13	16	16	13	8	5	2	1	0	0

a



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	8	5	2	1	1	0	0	0	0	0	0	0	0	1	2	4
1	5	2	1	0	0	0	0	0	0	0	0	0	0	0	1	1
2	2	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0
3	1	0	0	0	0	0	1	2	4	4	2	1	0	0	0	0
4	0	0	0	0	1	1	2	5	8	8	5	2	1	0	0	0
5	0	0	0	0	1	4	5	8	13	13	8	5	2	1	0	1
6	0	0	0	1	2	5	10	13	18	18	13	8	5	2	1	2
7	0	0	0	1	4	8	13	20	25	25	18	13	8	5	4	5
8	0	0	0	1	4	9	16	25	34	32	25	18	13	10	9	9
9	0	0	0	1	4	8	13	20	25	25	18	13	8	5	4	4
10	0	0	0	1	2	5	10	13	16	16	13	8	5	2	1	1
11	0	0	0	0	1	4	5	8	9	9	8	5	2	1	0	0
12	0	0	0	0	1	1	2	4	4	4	4	2	1	0	0	0
13	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
14	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	5	2	1	0	0	0	0	0	0	0	0	0	0	0	1	2

b

Рис. 17 Таблицы весов для печатных символов «*A*» (*a*) и «*C*» (*b*) с размерностью 16:16 для нейронов нулевого слоя

На рис. 17 приведены таблица весов для эталонных печатных символов «*A*» (Рис. 17*a*) и «*C*» (Рис. 17*b*) с размерностью 16 строк на 16 столбцов. Данные таблицы соответствуют таблицам весов нейронов нулевого слоя. Из рисунка можно также видеть, что активные ячейки таблиц имеют значение весов равные *0*. Данные таблицы весов были вычислены с использованием метрической меры близости - квадрата евклидового расстояния с использованием выше приведенного алгоритма..

В параграфе 1.19 на основе реализованного программного модуля выполняется оценка времени вычисления таблицы весов [26]. Время вычисления таблиц весов T для всех нейронов нулевого слоя, определится как $T = n \times t_{\text{ср}}$, где $t_{\text{ср}}$ - среднее время вычисления одной таблицы весов, n – количество нейронов первого или второго слоя. К примеру, если мы исходя из рис. 17, определяем среднее время вычисления таблицы весов как $t_{ср} = (0{,}078 + 0{,}093)/2 = 0{,}0855$ *с.*, то для задачи распознавания алфавитных символов с количеством

распознаваемых образов равным 30, в котором для каждого образа используется в среднем по 3 эталона (что в целом составляет 90 эталонов), понадобится время вычисления $T = 90 \times t_{\text{ср}} \approx 7\ c.$ для нейронной сети с нулевым слоем, и $T = 90 \times 89 \times t_{\text{ср}} = 6{,}23$ *мин.* для архитектуры расширенной нейронной сети без нулевого слоя и $T = 90 \times 89 \times t_{\text{ср}}/2 \approx 3$ *мин.* для сжатой схемы нейронной сети [8, 15].

Нужно отметить, что при обучении нейронной сети классическими алгоритмами требуется время значительно большее, которое измеряется часами, а иногда и сутками и неделями. Таким образом, вычисление значение весов требует сравнительно не значительного времени, при этом создается уже работающая нейронная сеть, которая может еще и до обучаться классическими алгоритмами.

В параграфе 1.20 приводится пример [27] вычисления значений весов с использованием реализованного алгоритма вычисления всех значений таблиц весов для задачи распознавания 6 образов рукописных букв {*A, B, E, M, N, Z*} с 8 эталонами, приведенными на рис. 18. На рис. 18 можно видеть, что образу символа «А» и «В» соответствует по два эталона, наименованные как: *A1, A2, B1, B2*. Для наглядности и простоты примера будем также использовать схему нейронной сети с нулевым слоем с расширенной схемой. Для нашей задачи *N*=8 (количество эталонов). Исходя из условия задачи количество нейронов нулевого слоя равно $n_0 = 8$, количество нейронов первого слоя $n_1 = 8*7=56$, количество нейронов второго слоя равно количеству эталонов $n_2 = 8$, количество нейронов третьего слоя равно количеству образов $n_3 = 6$. Размерность таблицы весов примем равной 8:8 и таким образом количество связей одного нейрона нулевого слоя будет равно 64. На рис. 18 приведены таблицы весов для каждого нейрона нулевого слоя, вычисленные описанным выше способом. Каждому нейрону нулевого слоя соответствует одна вычисленная таблицы весов на рис. 18 согласно также их расположению на рис. 18. Функции состояний нейронов нулевого слоя (Рис. 20) определяются путем перемножения таблиц весов с бинарной матрицей

входного распознаваемого элемента (Рис. 19*ab*). Функции состояний нейронов первого слоя приведены в табл. 3.

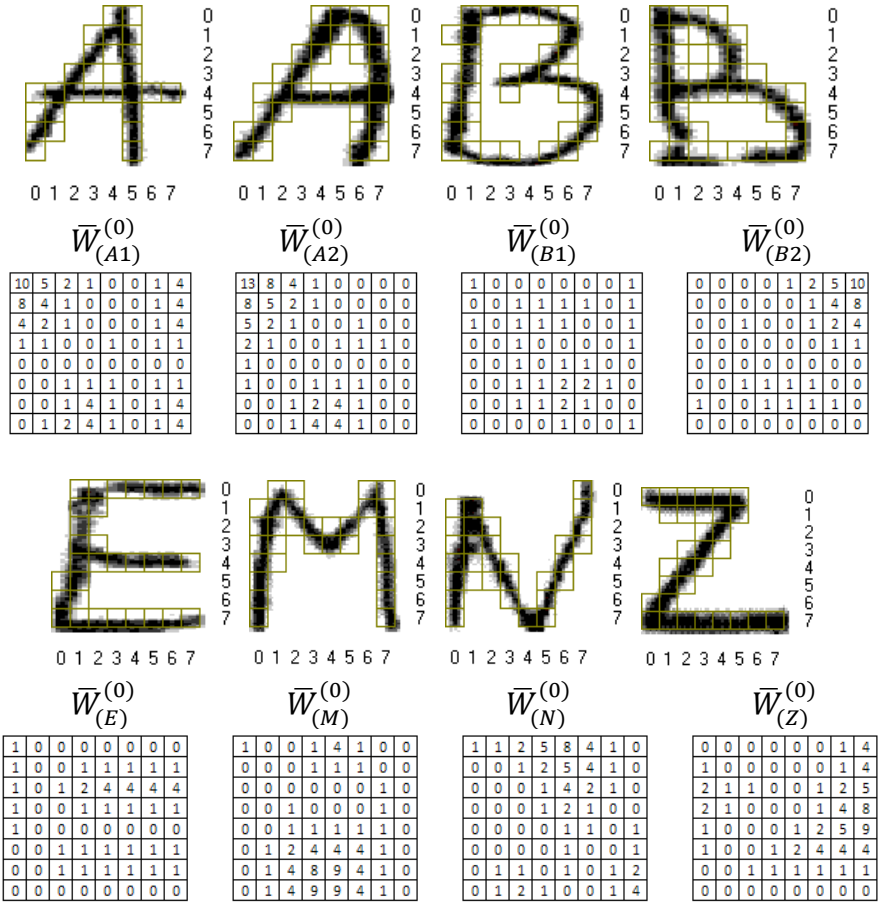
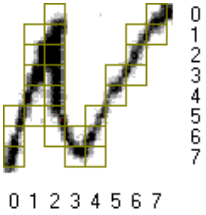


Рис. 18 Эталонные символы {*A1, A2, B1, B2, E, M, N, Z*} с размерностью матрицы разбиения изображений 8:8 и соответствующие им таблицы весов нейронов нулевого слоя.



	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	1
1	0	1	1	0	0	0	1	1
2	0	1	1	0	0	0	1	0
3	0	1	1	0	0	1	1	0
4	0	1	1	0	0	1	0	0
5	1	1	1	0	1	0	0	0
6	1	0	1	1	1	0	0	0
7	1	0	0	1	1	0	0	0

a *b*

Рис. 19 (*a*) Тестируемый рукописный символ «*N*», (*b*) Бинарная матрица тестируемого символа $\bar{X}$.

$\bar{X}\bar{W}_{(A1)}^{(0)}$

0	0	2	0	0	0	4
0	4	1	0	0	1	4
0	2	1	0	0	1	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	0	1	4	1	0	0
0	0	0	4	1	0	0

$Sw_1^{(0)} = 35$

$\bar{X}\bar{W}_{(A2)}^{(0)}$

0	0	4	0	0	0	0
0	5	2	0	0	0	0
0	2	1	0	0	0	0
0	1	0	0	0	1	1
0	0	0	0	0	0	0
1	0	0	0	1	0	0
0	0	1	2	4	0	0
0	0	0	4	4	0	0

$Sw_2^{(0)} = 34$

$\bar{X}\bar{W}_{(B1)}^{(0)}$

0	0	0	0	0	0	1
0	0	1	0	0	0	1
0	0	1	0	0	0	0
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	2	0	0
0	0	1	1	2	0	0
0	0	0	0	1	0	0

$Sw_3^{(0)} = 15$

$\bar{X}\bar{W}_{(B2)}^{(0)}$

0	0	0	0	0	0	0	10
0	0	0	0	0	0	4	8
0	0	1	0	0	0	2	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
1	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

$Sw_4^{(0)} = 31$

$\bar{X}\bar{W}_{(E)}^{(0)}$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
0	0	1	0	0	0	4	0
0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0

$Sw_5^{(0)} = 14$

$\bar{X}\bar{W}_{(M)}^{(0)}$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0
0	1	2	0	4	0	0	0
0	0	4	8	9	0	0	0
0	0	0	9	9	0	0	0

$Sw_6^{(0)} = 51$

$\bar{X}\bar{W}_{(N)}^{(0)}$

0	0	2	0	0	0	0	0
0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0
0	0	0	1	0	0	0	0

$Sw_7^{(0)} = 11$

$\bar{X}\bar{W}_{(Z)}^{(0)}$

0	0	0	0	0	0	0	4
0	0	0	0	0	0	1	4
0	1	1	0	0	0	2	0
0	1	0	0	0	1	4	0
0	0	0	0	0	2	0	0
1	0	0	0	2	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0

$Sw_8^{(0)} = 27$

Рис. 20 Таблицы значений произведений бинарной матрицы $\bar{X}$ тестируемого (входного) символа с таблицами весов нейронов нулевого слоя $\overline{W}^{(0)}$ и значения функций состояния $Sw_i^{(0)}$ нейронов нулевого слоя.

Табл. 3 Значения функций состояний $Sw_{i,j}^{(1)}$ нейронов первого слоя.

				A1	A2	B1	B2	E	M	N	Z
			j	1	2	3	4	5	6	7	8
			$Sw_j^{(0)}$	35	34	15	31	14	51	11	27
	i	$Sw_i^{(0)}$		$Sw_{i,j}^{(1)} = Sw_i^{(0)} - Sw_j^{(0)}$							
A1	1	35			1	20	4	21	-16	24	8
A2	2	34		-1		19	3	20	-17	23	7
B1	3	15		-20	-19		-16	1	-36	4	-12
B2	4	31		-4	-3	16		17	-20	20	4
E	5	14		-21	-20	-1	-17		-37	3	-13
M	6	51		16	17	36	20	37		40	24
N	7	11		-24	-23	-4	-20	-3	-40		-16
Z	8	27		-8	-7	12	-4	13	-24	16	

Табл. 4 Значения выходов нейронов первого, второго и третьего слоя.

			A1	A2	B1	B2	E	M	N	Z		Второй слой		Третий слой	
	j		1	2	3	4	5	6	7	8					
	i		$f(Sw_{i,j}^{(1)})$									$Sw_k^{(2)}$	$f(Sw_k^{(2)})$	$Sw_{\text{обр}}^{(3)}$	Y_{out}
A1	1			0	0	0	0	1	0	0		1	0	0+0=0	0
A2	2		1		0	0	0	1	0	0		2	0		
B1	3		1	1		1	0	1	0	1		5	0	0+0=0	0
B2	4		1	1	0		0	1	0	0		3	0		
E	5		1	1	1	1		1	0	1		6	0	0	0
M	6		0	0	0	0	0		0	0		0	0	0	0
N	**7**		**1**	**1**	**1**	**1**	**1**	**1**		**1**		**7**	**1**	**1**	**1**
Z	8		1	1	0	1	0	1	0			4	0	0	0

В табл. 4 можно видеть, что уже на этом этапе 7-ой выход второго слоя, соответствующий эталону «*N*», является активным. В третьем слое нейронной сети выполняется объединение нейронов одного образа в одном нейроне. Для данного примера третий слой необходим для первого и второго образов «А» и «В», поскольку эти образы имеют по два эталона

и их выходы объединяются в нейронах третьего слоя. Функция состояния $Sn_k^{(3)}$ и $f\left(Sn_k^{(3)}\right)$ активации нейрона третьего слоя определяются по формулам (10), (11).

Из табл. 4 можно видеть, что в итоге для входного символа на рис. 19а активизируется 5-ый выход нейронной сети, соответствующий образу «*N*».

В параграфе 1.21. приводятся выводы по первой главе.

В главе II исследуются возможности предложенных схем нейронной сети для возможного применения в многозадачных реализациях [14, 15]. Рассматриваются преимущества предложенных архитектур нейронных сетей в многозадачных применениях.

В параграфе 2.1. рассматривается целесообразность создания в многозадачном применении единого классификатора [14] с использованием трехслойной нейронной сети с нулевым слоем.

В параграфе 2.2. рассматривается целесообразность создания в многозадачном применении единого классификатора с использованием четырехслойной [14] нейронной сети с нулевым слоем.

В параграфе 2.3. приводится схема применения одной нейронной сети к разным задачам для случая, когда известно принадлежность входного объекта к той или иной задачи [15]. На рис. 21 приведена общая схема применения нейронной сети в многозадачном использовании с различно определенными классификаторами, для случая, если параметры для каждой применяемой задачи, такие как количества используемых эталонов, количества используемых признаков образца, количества распознаваемых образов - будут соответствовать параметрам применяемой нейронной сети: количеству нейронов нулевого слоя, размерности таблиц весов, количеству выходов нейронной сети [8, 15].

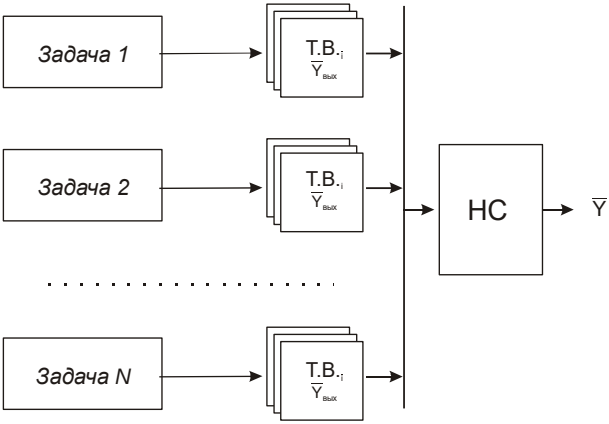


Рис. 21 Схема многозадачного применения нейронной сети.

В данном параграфе также исследуются схемы реализации нейронной сети в многозадачном применении для случаев, когда входные начальные параметры, используемые в задачах распознавания, такие как: количество эталонов, размерность таблицы весов – отличаются от задачи к задаче. В данном параграфе рассмотрены случаи применения задач к нейронной сети, при которых параметры сети не сходятся с параметрами применяемой задачи. Например:

1. Если для какой то задачи, размерность таблицы весов (m, k) меньше, чем размерность таблицы весов (M, K) нейронной сети ($m<M$, $k<K$) (Рис. 22), то в этом случае не используемые столбцы и строки таблицы можно обнулить (Рис. 22), что согласно выражению (28) сохранит целостность и логику работы нейрона нулевого слоя:

$$Sn_0 = \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij}w_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{k} x_{ij}w_{ij} + \sum_{i=m+1}^{M}\sum_{j=k+1}^{K} x_{ij}w_{ij} = \\ = \sum_{i=1}^{m}\sum_{j=1}^{k} x_{ij}w_{ij} + \sum_{i=m+1}^{M}\sum_{j=k+1}^{K} (x_{ij}\times 0) = \sum_{i=1}^{m}\sum_{j=1}^{k} x_{ij}w_{ij} \qquad (28)$$

где m, k – количество столбцов и строк таблицы весов задачи (на рис. 22*a* m = 3, k = 4); M, K – количество столбцов и строк таблицы весов нейронной сети (на рис. 22*b* M = 5, K = 7).

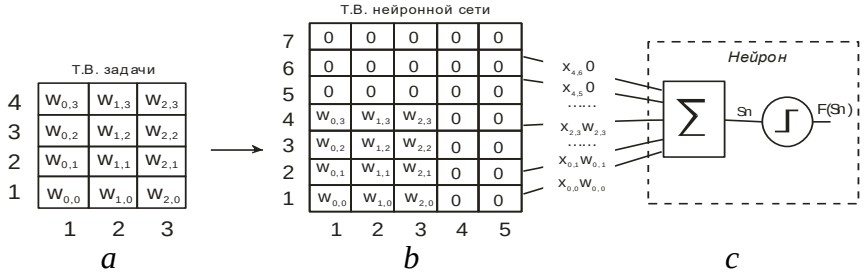


Рис. 22 Схема применения таблицы весов задачи с размерностью меньше количества связей нейрона.

2. В случае, если количество эталонов, для какой то задачи меньше, чем количество нейронов нулевого слоя, то для сохранения целостности работы сети для данной задачи, для не задействованных нейронов нулевого слоя создается таблица весов «удаленного образа» (Т.В.у.о.), характеризуемого значительной удаленностью от всех остальных эталонов. При этом образец «удаленного образа» создается исходя из условия, при котором сравнение между двумя любыми эталонами, с использованием характеристики близости используемого в метрическом методе распознавания, было меньше, чем идентичное сравнение любого эталона с образцом «удаленного образа» [14]. Для соблюдения этого условия достаточно выполнение условия (29), при котором все значения весов Т.В.у.о. принимают одно значение (w_r) большее, чем значение максимального веса (w_{max}), определяемого из набора всех значений весов ($w_{ij,k}$) в наборе реальных таблиц весов.

$$w_{ij,k} \le w_{\max} \le w_r \qquad (29)$$

При этом значение w_r может определяться как:

$$w_r = Lw_{\max}, \qquad (30)$$

где $L \in [1,\infty]$– коэффициент удаленности «удаленного образа». Можно при этом показать, что исходя из (29), (30) на выходе *i*-го нейрона второго слоя нейронной сети, соответствующему «удаленному образу», значение будет всегда равно нулю. Для этого сначала оценим значение нейрона первого слоя (Sn_{1,k_r}),

выполняющего сравнение эталонного образца с образцом «удаленного образа»:

$$Sn_{1,k_r} = \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} w_{ij,k} - \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} w_r \le \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} w_{\max} - \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} w_r =$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{K} \left(x_{ij} w_{\max} - x_{ij} w_r \right) = \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} \left(w_{\max} - w_r \right) = \quad (31)$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{K} x_{ij} \left(w_{\max} - L w_{\max} \right) < 0$$

Отсюда, функция активации для данного нейрона $f(Sn_{1,k_r})$ равна *1*. В этом случае значение $Sn_{2,r}$ для нейрона второго слоя соответствующему «удаленному образу», определится для случая α=1:

$$Sn_{2,r} = \sum_{i=1}^{r-1} \overline{f}(Sn_{1,i_r}) + \sum_{i=r+1}^{N} f(Sn_{1,r_i}) =$$
$$\sum_{i=1}^{r-1,(i\neq k)} \overline{f}(Sn_{1,i_r}) + \overline{f}(Sn_{1,k_r}) + \sum_{i=r+1}^{N} f(Sn_{1,r_i}) = \quad (32)$$
$$= \sum_{i=1}^{r-1,(i\neq k)} \overline{f}(Sn_{1,i_r}) + \overline{1} + \sum_{i=r+1}^{N} f(Sn_{1,r_i}) < N-1$$

где r – номер нейрона соответствующего «удаленному образу» в нулевом слое; N –количество образов (эталонов) нейронной сети. Таким образом получим выходное значение нейрона второго слоя для «удаленного образа» $f(Sn_{2,r}) = 0$. То есть добавление "удаленного образа" не меняет логику работы сети.

В параграфе 2.4. приводятся схемы применения нейронной сети в многозадачном применении для случая, когда принадлежность входных объектов к той или иной задачи не известно. При этом в схему нейронной сети многозадачного применения добавляется задача выделения классификатора под задачи. Для этого предлагается добавить в нейронную сеть дополнительную задачу (классификатор) (Рис. 23). При этом для определения данного классификатора каждая задача представляется, как отдельный образ (класс), для которого определяется эталон или эталоны класса, например по приведенному в [15] алгоритму отбора. На основе эталонов

класса создаются таблицы весов классов и одна таблица «удаленного образа» для данной задачи. Процедура распознавания входного элемента выполняется по схеме рис. 23. На первом этапе выполняется определение необходимого класса, которой соответствует распознаваемый объект (Рис. 23(1)), для чего на входы сети подаются таблицы весов классов, а на не использованные нейроны нулевого слоя подаются значения таблицы весов «удаленного образа». При этом номер активного выхода нейронной сети, определяющий номер задачи, которой принадлежит объект, активизирует классификатор соответствующей задачи посредством обратной связи (Рис. 23(2)). Таблицы весов выбранной задачи поступают на входы нейронной сети, а выходной эталонный набор $Y_{вых}$ для этой задачи поступает на выход нейронной сети (Рис. 23(3)), после чего выполняется распознавание объекта в рамках выделенной задачи.

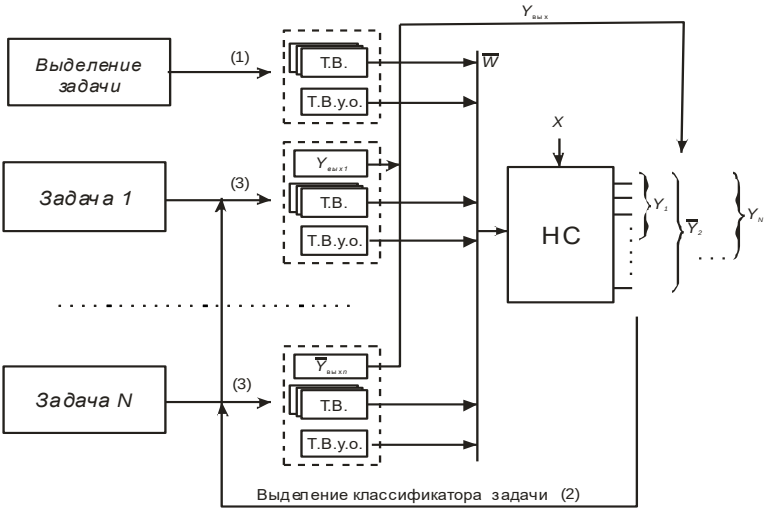


Рис. 23 Схема работы сети с предварительным выделением классификатора задачи.

В параграфе 2.5. предложен алгоритм последовательного применения эталонов к нейронной сети на основе метрического метода распознавания. Приводится схема и алгоритм

использования нейронной сети в многозадачном применении для случаев, когда в задачи количество эталонов больше, чем это позволяют ресурсы нейронной сети, то есть больше количества нейронов нулевого слоя.

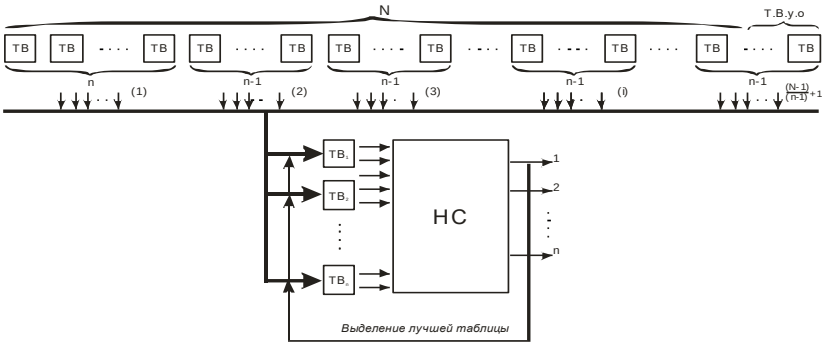


Рис. 24 Схема последовательного перебора эталонов в сети.

При выполнении условия $n < N_{et}$, где n количество нейронов нулевого слоя, N_{et} количество используемых эталонов, первоначально на сеть подаются таблицы весов первых n эталонов. Результаты на выходе нейронной сети посредством обратных связей (Рис. 24) выделяют соответствующий активному выходу i-ую таблицу весов ($Т.В._i$) наиболее ближайшего эталона, значение весов которых повторно подаются на входы i-го нейрона нулевого слоя. При этом на оставшиеся $(n–1)$ нейрона нулевого слоя подаются таблицы весов следующих $(n–1)$ эталонов. Процедура продолжается аналогичным образом до тех пор, пока не будут использованы все (N_{et}) таблицы весов эталонов. При этом если на каком то этапе алгоритма последовательного перебора (например на последнем, рис. 24), какие то нейроны нулевого слоя остаются не использованными, то на не задействованные нейроны нулевого слоя подается таблица весов «удаленного образа» определенного для этой задачи.

В параграфах 2.6. и 2.7. приводятся простейшие примеры реализации многозадачного применения нейронной сети на

основе метрического метода распознавания. Например на рис. 25 приведена схема применения двух простейших задач к одной сети. В каждой задаче определяются два образа. Для первой задачи $Y_{вых}$=*{кривая1, кривая2}*, для второй $Y_{вых}$={*7, 2*}.

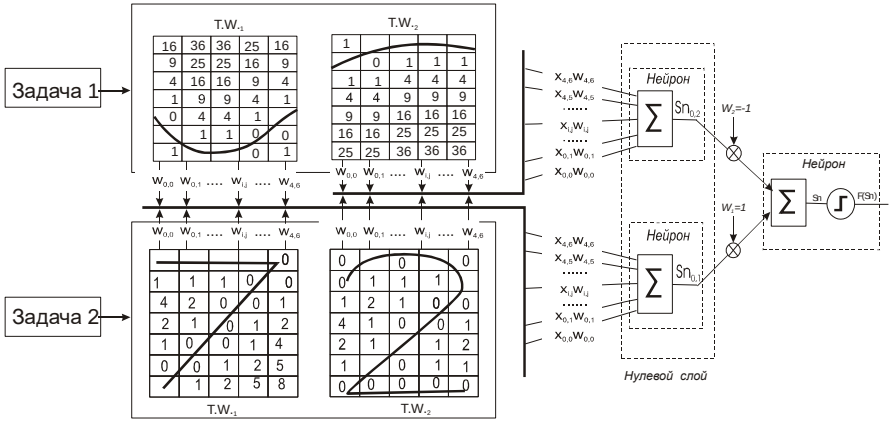


Рис. 25 Схема применения нейронной сети к двум задачам с двумя эталонами.

В параграфе 2.8. приводятся выводы, касательные главы II.

В главе III рассматриваются схемы и алгоритмы применения нейронных сетей на основе метрических методов распознавания в задачах принятия решений с нечеткими выводами [9].

В параграфе 3.1. предлагается алгоритм преобразования задачи нечеткого вывода в задачу распознавания кривых. Для этого предлагается схема преобразование нечетких множеств и нечетких правил задачи нечеткого вывода в набор кривых.

На рис. 26(*a,b,c*) приведены примеры нечетких множеств x_i определенные на диапазонах Δx_i. При этом если нечеткие множества определены на отдельных элементах множества, то в этом случае определенные наборы выделенных элементов множеств можно представить в виде отдельных эталонных наборов $\overline{X}$.

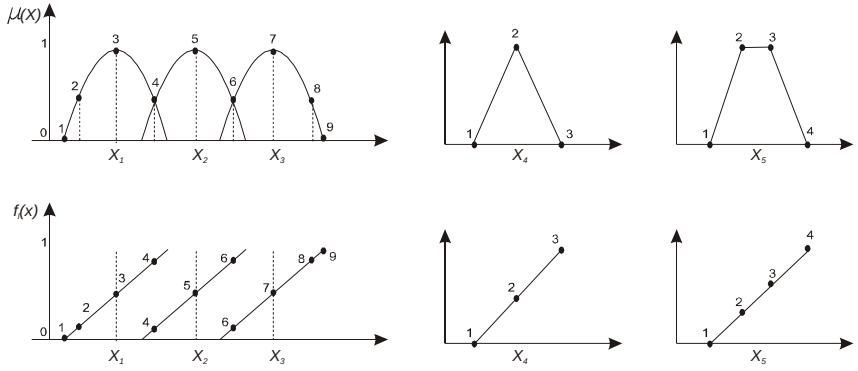


Рис. 26 Преобразование множеств в линейную форму.

Приведенное множество на рис. 26(*a*), состоит из трех нечетких подмножеств X_1, X_2, X_3, определенных на 9-ти выделенных элементах множества. Каждому выделенному элементу подмножества соответствует значение принадлежности $\mu(x)$. При этом если нечеткое подмножество X_i определено на некотором диапазоне значений $[x_{i,1}, x_{i,2}]$, то линейные значения $f_l(x_{i,j})$ выделенных элементов $x_{i,j}$ определятся из выражения:

$$f_l(x_{i,j}) = \frac{1}{x_{i,2} - x_{i,1}} x_{i,j} + \frac{1}{x_{i,2} - x_{i,1}} x_{i,1} \qquad (33)$$

где $f_l(x_i)$ - линейная функция i-го подмножества.

Если каждое нечеткое подмножества на рис. 26(а) представить в виде одного отсчета кривой, то получим кривую, состоящую из 3-ех отсчетов. При этом для определения количества возможных кривых, назовем два множества (подмножества) зависимыми, если каждому элементу одного множества может быть сопоставлен только один элемент другого множества (условие однозначного отображения) и не зависимыми, если каждому элементу одного множества может быть сопоставлен любой элемент другого множества.

Например, приведенные на рис. 26*a* подмножества являются зависимыми, поскольку являются термами одного

множества. Для таких множеств количество кривых будет соответствовать количеству выделенных элементов множества. На рис. 27 приведены 9 возможных кривых полученных на основе 9-ти линейных значений $f_l(x_{i,j})$ элементов множества на рис. 26*а*.

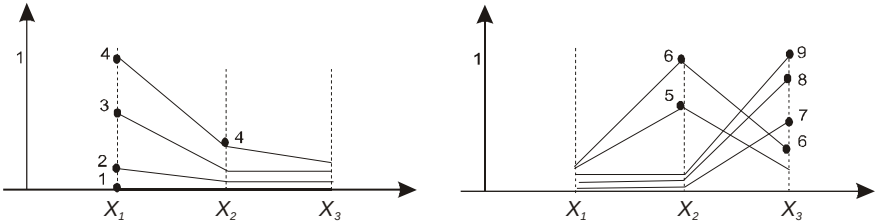


Рис. 27 Наборы кривых для подмножеств *X1, X2, X3*.

Каждой эталонной кривой на выходе системы распознавания может соответствовать одно или несколько выходных значений y_i (Рис. 28).

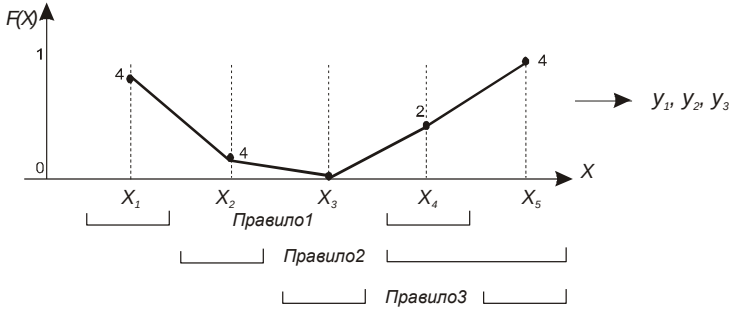


Рис. 28 Пример одной эталонной кривой.

Рассмотрим пример возможного набора нечетких правил применительно к нечетким множествам, приведенным на рис. 26:

Правило 1: Если x равен X_1 и x равен X_4 то y равен Y_1

Правило 2: Если x равен X_2 и x равен X_4 или x равен X_5 то y равен Y_2

Правило 3: Если x равен X_3 или x равен X_5 то y равен Y_3

На рис. 28 можно видеть, что эталонная кривая объединяет приведенные 3 нечетких правила, при этом на выходе системы распознавания эталонной кривой соответствует три эталонных выходных значения y_1, y_2, y_3, соответствующие множествам Y_1, Y_2, Y_3 (Рис. 28). Аналогичным образом полученные кривые могут включать в себя и возможные другие правила, составленные на основе множеств, на основе которых кривые создавались.

В параграфе 3.2. приводится схема применения нейронной сети на основе метрических методов распознавания к задаче распознавания полученного набора кривых и принятия на основе предлагаемых нейронных сетей окончательного решения нечеткого вывода (Рис. 29, 31).

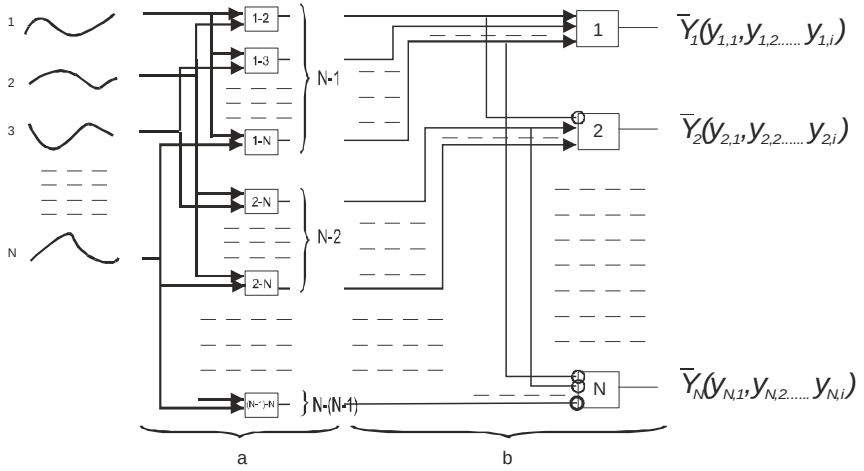


Рис. 29 Нейронная сеть для *N* образов.

Каждому *i*–му выходу нейронной сети на рис. 29 будет соответствовать выходной вектор $\overline{Y}_i$, значения которого будут активизированы, если значение соответствующего выхода нейронной сети будет равно 1. В качестве значений выходного вектора нейронной сети на рис. 29 могут быть заданы как значения линейных функций $f_l(y_{i,j})$, для всех выходных

множеств, так и непосредственно значения самих выходных множеств $\overline{Y_i}$, определенных согласно применяемому алгоритму нечеткого вывода, либо из выражения (33) если известны линейные значения функции $f_l(y_{i,j})$:

$$y_{i,j} = (y_{i,2} - y_{i,1})\left(f_l(y_{i,j}) - \frac{1}{y_{i,2} - y_{i,1}} y_{i,1} \right) \qquad (34)$$

Для получения более точных значений Y_i необходимо определение промежуточных результатов выходных значений, значение которых можно определить на основе значений коэффициентов близости S_k.

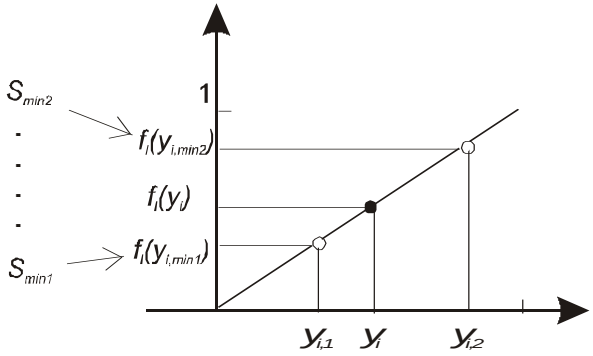


Рис. 30 Определение промежуточных значений.

Для этого по отношению к тестируемой кривой определяются значения S_k для ближайших двух эталонных кривых, $S_{\min 1}$ и $S_{\min 2}$ (Рис. 30). При этом выходное значение линейной функции $f_l(y_i)$, будет определяться из выражения

$$\frac{f_l(y_{i,\min 1}) - f_l(y_i)}{f_l(y_i) - f_l(y_{i,\min 2})} = \frac{\sqrt{S_{\min 1}/N}}{\sqrt{S_{\min 2}/N}}, \qquad (35)$$

где $f_l(y_i)$ - итоговое промежуточное значение линейной функции для *i*-той выходной переменной, $f_l(y_{i,\min 1})$, $f_l(y_{i,\min 2})$ - значения линейных функций для *i*-го выходного значения соответствующие ближайшим двум эталонным кривым на

выходах нейронной сети (Рис. 29), N - количество отсчетов кривых. Поскольку функция f_l является линейной по отношению к своей переменной то в выражении (35) можно заменить значение функции f_l на значение переменных:

$$\frac{y_{i,\min 1} - y_i}{y_i - y_{i,\min 2}} = \frac{\sqrt{S_{\min 1}/N}}{\sqrt{S_{\min 2}/N}}, \text{ отсюда}$$

$$y_i = \frac{y_{i,\min 1}\sqrt{S_{\min 2}} + y_{i,\min 2}\sqrt{S_{\min 1}}}{\sqrt{S_{\min 1}} + \sqrt{S_{\min 2}}} \qquad (36)$$

Выражение (36) определяет выходные значения для всех выходных значений нечетких множеств то есть выполняется дефазифицирование итоговых выходных значений. Для реализации выражения (36) необходимо определить $S_{\min 1}$, $S_{\min 2}$, значение которых можно получить непосредственно из нейронной сети. Для этого в нейронную сеть (Рис. 29) добавляется дополнительный нулевой слой (Рис. 31*а*), состоящий из линейных нейронов, функции активации которых равны значениям самих нейронов (15).

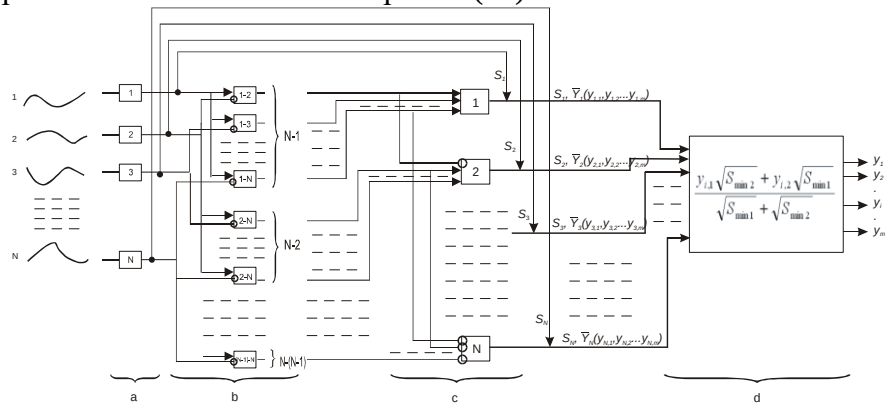


Рис. 31 Нейронная сеть и блок дефазификации для *N* образов.

В параграфе 3.3. рассматриваются возможности упрощения и минимизации схемы на рис. 29, 30 с целью уменьшения количества нейронов, размерности таблиц весов. С этой целью предлагается разбиение набора кривых на диапазоны кривых определяемым по нечетким правилам и таким образом преобразования единой нейронной сети на основе метрического метода распознавания в многоблочную структуру нейронной сети (Рис. 32).

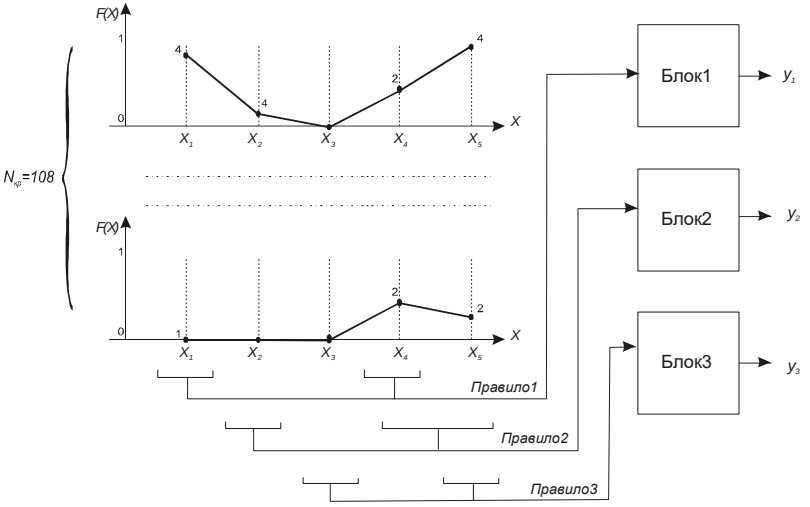


Рис. 32 Разбиение сети на отдельные блоки.

С целью дальнейшей минимизации нейронной сети рассматривается также применение одна блочной нейронной сети в многозадачном применении для всех нечетких правил.

В параграфе 3.4. предлагается алгоритмы отбора мнений экспертов применительно к нейронной сети на основе метрического метода распознавания.

В параграфе 3.5. приводятся итоговые выводы для главы III.

В главе IV выполняется преобразование нейронной сети на основе метрического метода распознавания в полно связный многослойный персептрон [18, 19]. С этой целью исследуются

закономерности определяющие распределения весовых и пороговых значений нейронной сети на основе метрических методов распознавания. В результате исследуются условия необходимые при выполнении распределений весовых и пороговых значений нейронов второго и третьего слоя для полно связной и не полно связной нейронной сети. Приводится обобщенный алгоритм вычисления весовых и пороговых значений полно связного многослойного персептрона на основе метрического метода распознавания. Показано, что нейронная сеть на основе метрических методов распознавания является частным случаем многослойного персептрона, и соответственно может до обучаться классическими алгоритмами обучения. При этом в отличии от обычных архитектур многослойного персептрона полученный многослойный персептрон сохраняют все преимущества нейронной сети на основе метрического метода распознавания.

В параграфе 4.1. приводится общая постановка задачи, а также обоснование целесообразности решения данной задачи.

В параграфе 4.2. исследуются обобщенные закономерности распределение значений весовых и пороговых значений для второго слоя нейронной сети на основе метрических методов распознавания. Было показано, что при распределении входных весовых $w_{k,j}^{(2)}$ и пороговых$H_k^{(2)}$значений k-го нейрона второго слоя для схемы не полно связной сети на рис. 2 должно выполнять следующее условие.

$$\sum_{\substack{j=1 \\ (j\neq k)}}^{N-1} w_{k,j}^{(2)} - w_{k,j}{}_{min}^{(2)} < H_k^{(2)} \leq \sum_{\substack{j=1 \\ (j\neq k)}}^{N-1} w_{k,j}^{(2)}, \qquad (37)$$

где $w_{k,j}{}_{min}^{(2)}$ минимальное входное значение $w_{k,j}^{(2)}$ k-го нейрона второго слоя.

Было показано также, что для выполнения условия функции активации (5) k-го нейрона второго слоя, необходимо также выполнения условий $w_{k,j}^{(2)} > 0$ для всех j входов нейрона второго слоя .

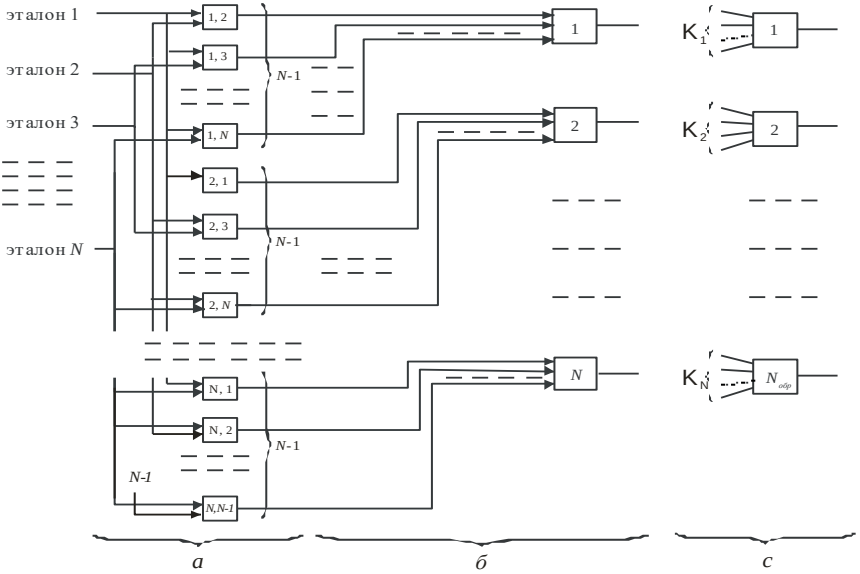


Рис. 33 Трехслойная нейронная сеть на основе метрических методов распознавания с не минимизированной архитектурой.

$$\begin{cases} f\left(Sn_k^{(2)}\right) = 1, \text{если } Sn_k^{(2)} \geq H_k^{(2)} \\ f\left(Sn_k^{(2)}\right) = 0, \text{если } Sn_k^{(2)} < H_k^{(2)} \end{cases} \quad (38)$$

где $Sn_k^{(2)}$ – функция состояния k-го нейрона второго слоя.

Было показано, что для случая преобразования нейронной сети на рис. 33 в полно связную нейронную сеть, значения состояния k-го нейрона второго слоя преобразуется в выражение:

$$Sn_k^{(2)} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ (j \neq i)}}^{N-1} \left(w_{i,j}^{(2)} f\left(Sn_{i,j}^{(1)}\right)\right), \quad (39)$$

где i, j определяют порядковые номера попарно разделяемых эталонов (Рис. 33*а*). При этом, было также показано, что для k-го нейрона второго слоя для всех добавленных связей ($i \neq k$), с целью соблюдения стабильности работы функции активации (5) необходимо также выполнения условий (40), (41):

$$\sum_{\substack{i=1,\\ i\neq k}}^{N-1}\sum_{\substack{j=1,\\ j\neq i,\\ w_{i,j}^{(2)}\geq 0}}^{N-1} w_{i,j}^{(2)} < H_k^{(2)} - \left(\sum_{\substack{j=1,\\ j\neq k}}^{N-1}\left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)}\right)\right) - w_{k,j\ min}^{(2)}\right), \quad (40)$$

$$\left|\sum_{\substack{i=1,\\ i\neq k}}^{N-1}\sum_{\substack{j=1,\\ j\neq i,\\ w_{i,j}^{(2)}<0}}^{N-1} w_{i,j}^{(2)}\right| \leq \sum_{\substack{j=1,\\ J\neq k}}^{N-1}\left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)}\right)\right) - H_k^{(2)}, \quad (41)$$

где левые части в выражениях (40), (41) положительные и отрицательные составляющие суммы всех $w_{i,j}^{(2)}$ для k-го нейрона второго слоя, для которых $i \neq k$. Не соблюдение условий (40), (41), будет означать возможность появлений ошибочных значений на выходе k-го нейрона второго слоя.

В параграфе 4.3. исследуются закономерности распределения весовых и пороговых значений третьего слоя нейронной сети. Показано, что для соблюдения условия функции активации третьего слоя (10):

$$\begin{cases} f\left(Sn_k^{(3)}\right) = 1, \text{если } Sn_k^{(3)} \geq H_k^{(3)} \\ f\left(Sn_k^{(3)}\right) = 0, \text{если } Sn_k^{(3)} < H_k^{(3)} \end{cases} \quad (42)$$

необходимо, что бы пороговое значение k-го нейрона третьего слоя $H_k^{(3)}$ для не полносвязной нейронной сети определялось в диапазоне (43):

$$0 < H_k^{(3)} \leq w_{k,j}^{(3)(\min)} \quad (43)$$

где $w_{k,j}^{(3)(\min)}$минимальное значение веса $w_{k,j}^{(3)}$ для k-го нейрона третьего слоя.

Было также показано, что для полно связной нейронной сети для стабильности работы третьего слоя, значения весов добавленных связей $w_{k,j}^{(3)}$ при $j \neq j_k$, должны определяться условиями:

$$\sum_{\substack{j=1,\\ j\neq j_k,\\ w_{k,j}^{(2)}\geq 0}}^{N} w_{k,j}^{(3)} < H_k^{(3)} \tag{44}$$

$$\left|\sum_{\substack{j=1,\\ j\neq j_k,\\ w_{k,j}^{(2)}<0}}^{N} w_{k,j}^{(3)}\right| \leq w_{k,j_k}^{(3)(min)} - H_k^{(3)}, \tag{45}$$

где j_k - j-ые входы k-го нейрона третьего слоя, соединяющиеся с нейронами второго слоя, выходы которых соответствуют эталонам k-го образа, $w_{k,j_k}^{(3)(min)}$ – минимальное значение веса для j_k связей. Левые части выражений (44, 45) - это положительные и отрицательные составляющие суммы весов для всех добавленных связей k-го нейрона третьего слоя ($j \neq j_k$).

В параграфе 4.4. на основе приведенных выше исследований был приведен общий алгоритм определения распределения пороговых и весовых значений для персептрона, созданного на основе метрического метода распознавания. В частности, предлагаемый алгоритм для определения весовых и пороговых значений второго слоя полно связной нейронной сети выглядит следующим образом:

1. На основе существующего эталонного набора строится сеть по схеме на рис. 33, в которую далее добавляются связи в результате чего сеть преобразуется в полно связный многослойный персептрон. При этом количество нейронов для первого слоя определяется по выражению (7), количество нейронов второго слоя определяется количеством эталонов, количество нейронов третьего слоя количеством распознаваемых образов.

2. Для k-го нейрона значения весов для всех добавленных связей $w_{i,j}^{(2)}$ (для которых $i \neq k$) выбираются случайным образом.

3. Из полученных значений $w_{i,j}^{(2)}$ определяются значения сумм S_1 и S_2 положительных и отрицательных составляющих, взятых по модулю:

$$S_1 = \sum_{\substack{i=1 \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} \geq 0}}^{N-1} w_{i,j}^{(2)}, S_2 = \left| \sum_{\substack{i=1 \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} < 0}}^{N-1} w_{i,j}^{(2)} \right| \quad (46)$$

4. Определяем минимальное значение $w_{k,j}^{(2)}{}_{min}$, по выражению:

$$w_{k,j}^{(2)}{}_{min} = S_1 + S_2 + \beta \quad (47)$$

где β – любое положительное число, $\beta \in (0, \infty)$. При этом выражение (47) определяется на основе сложений правых и левых частей выражений (44) и (45) и приведением неравенства к равенству:

$$\sum_{\substack{i=1, \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} \geq 0}}^{N-1} w_{i,j}^{(2)} + \left| \sum_{\substack{i=1,, \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} < 0}}^{N-1} w_{i,j}^{(2)} \right| << H_k^{(2)} -$$

$$\left(\sum_{\substack{j=1, \\ j\neq k}}^{N-1} \left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)} \right) \right) - w_{k,j}^{(2)}{}_{min} \right) + + \sum_{\substack{j=1, \\ J\neq k}}^{N-1} \left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)} \right) \right) -$$

$$H_k^{(2)} \quad (48)$$

$$\sum_{\substack{i=1 \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} \geq 0}}^{N-1} w_{i,j}^{(2)} + \left| \sum_{\substack{i=1, \\ i\neq k}}^{N-1} \sum_{\substack{j=1, \\ j\neq i, \\ w_{i,j}^{(2)} < 0}}^{N-1} w_{i,j}^{(2)} \right| < w_{k,j}^{(2)}{}_{min} \quad (49)$$

$$\sum_{\substack{i=1,\\ i\neq k}}^{N-1}\sum_{\substack{j=1,\\ j\neq i,\\ w_{i,j}^{(2)}\geq 0}}^{N-1} w_{i,j}^{(2)} + \left|\sum_{\substack{i=1,\\ i\neq k}}^{N-1}\sum_{\substack{j=1,\\ j\neq i,\\ w_{i,j}^{(2)}<0}}^{N-1} w_{i,j}^{(2)}\right| + \beta = w_{k,j}^{(2)}{}_{min} \quad (50)$$

5. Выбираем случайным образом остальные значения $w_{k,j}^{(2)}$, при этом соблюдая условие $w_{k,j}^{(2)} \geq w_{k,j}^{(2)}{}_{min}$, а также определяем сумму S_k всех $w_{k,j}^{(2)}$.

$$S_k = \sum_{\substack{j=1,\\ j\neq k}}^{N-1}\left(w_{k,j}^{(2)} f\left(Sn_{k,j}^{(1)}\right)\right) \quad (51)$$

6. Определяем значения порога $H_k^{(2)}$ для этого нейрона:

$$H_k^{(2)} = \left(S_k - w_{k,j}^{(2)}{}_{min}\right) + S_1 + c \quad (52)$$

где $c \in (0;\ \beta]$

7. Переходим на следующий k+1 нейрон и продолжаем алгоритм с пункта 2.

Для третьего слоя алгоритм определения значений $w_{k,j}^{(3)}$ и $H_k^{(3)}$ будет отличаться тем, что в пункте 4 определится минимальное значение $w_{k,j_k}^{(3)(min)}$, как

$$w_{k,j_k}^{(3)(min)} = S_1 + S_2 + \beta\ , \quad (53)$$

где

$$S_1 = \sum_{\substack{j=1,\\ j\neq j_k,\\ w_{k,j}^{(2)}\geq 0}}^{N} w_{k,j}^{(3)}\ , S_2 = \left|\sum_{\substack{j=1,\\ j\neq j_k,\\ w_{k,j}^{(2)}<0}}^{N} w_{k,j}^{(3)}\right|, \quad (54)$$

а пороговое значение $H_k^{(3)}$ для k-го нейрона третьего слоя определится из выражения (55), как

$$H_k^{(3)} = S_1 + c \quad (55)$$

где $c \in (0;\ \beta]$.

В параграфе 4.5. приводится математическое обоснование распределений пороговых и весовых значений на примере использования алгоритма обучения по методу Видроу Хоффа. На основе данного алгоритма обучения показано, что при обучении полно связного многослойного персептрона распределение значений весовых и пороговых значений стремится к выполнению условий (40), (41), (43), (44), (45), (49).

В параграфе 4.6. приводятся простейший пример, демонстрирующий создание многослойного персептрона с полной системой связей и с пороговой функцией активацией с вычисляемыми параметрами. В качестве простейшего примера приводится задача распознавания двух рукописных графических образов {4, 3}, состоящих, из набора, трех эталонов, приведенных на рис. 34.

Эталон 1 Эталон 2 Эталон 3

Рис. 34 Пример набора эталонов для двух образов.

В примере использовался общий алгоритм определения распределения пороговых и весовых значений для второго (табл. 5) и третьего слоя (табл. 6) многослойного персептрона, созданного на основе метрического метода распознавания.

Табл. 5. Значение весов $w_{i,j}^{(2)}$ и параметры алгоритма для второго слоя.

$w_{i,j}^{(2)}$ для k-го нейрона								Параметры алгоритма для 2-го слоя						
i,j / k	1, 2	1, 3	2, 1	2, 3	3, 1	3, 2		S_1	S_2	β	$w_{k,j}^{(2)}{}_{min}$	S_k	c	$H_k^{(2)}$
1	19	15	-2	-2	3	4		7	4	4	15	34	3	29
2	-4	2	11	14	0	3		5	4	2	11	25	1	20
3	-5	3	2	5	22	17		10	5	2	17	39	1	33

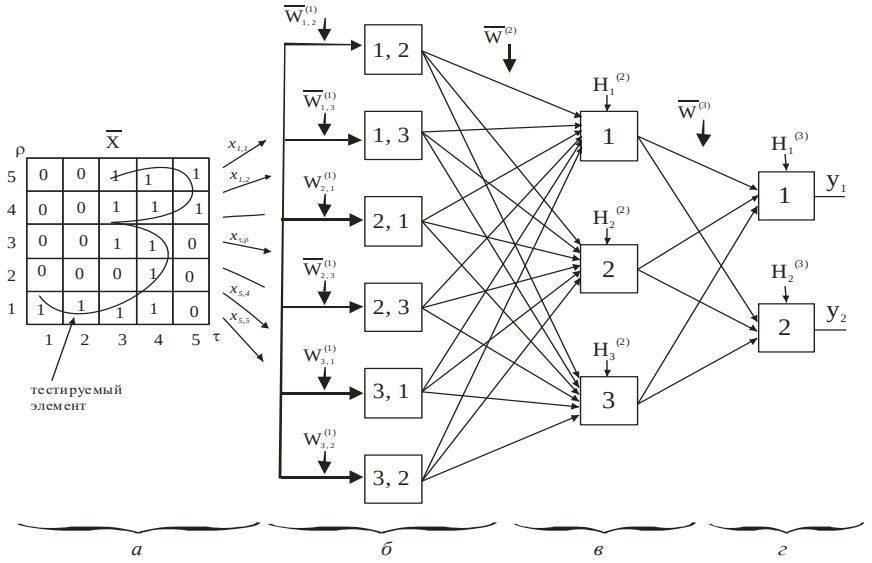


Рис. 35 Схема распознавания с использованием трехслойной нейронной сети, (а) тестируемый элемент $\overline{X}$ (б) первый слой, (в) второй слой, (с) третий слой.

Табл. 6. Значение весов $w_{k,j}^{(3)}$ и параметры алгоритма для третьего слоя.

$w_{k,j}^{(3)}$ для k-го нейрона					Параметры алгоритма для 3-го слоя						
k \ j	1	2	3		S_1	S_2	β	$w_{k,j_k}^{(3)(min)}$	S_k	c	$H_k^{(3)}$
1	*7*	2	5		2	0	3	5	12	2	4
2	1	9	-4		1	4	4	9	9	1	2

Был приведен также пример (табл. 7) распознавания тестируемого символа «3» (Рис. 35), который продемонстрировал, что при данных вычисленных параметрах, нейронная сеть работает правильно и идентифицирует тестируемый символ рукописной цифры «3».

Табл. 7. Значений нейронов и функций активаций для элемента $\overline{X}$ на рис. 35.

первый слой				второй слой				третий слой				$Y_{вых}$
№ (i, j)	$Sn_{i,j}^{(1)}$	$f(Sn_{i,j}^{(1)})$		№ k	$Sn_k^{(2)}$	$f(Sn_k^{(2)})$		№ k	$Sn_k^{(3)}$	$y_k = f(Sn_k^{(3)})$		
1, 2	-18	≥ 0	0	1	-1	≥29	0	1	2	≥4	0	-
1, 3	-14	≥ 0	0	2	25	≥20	1	2	9	≥2	1	{3}
2, 1	18	≥ 0	1	3	29	≥33	0					
2, 3	3	≥ 0	1									
3, 1	14	≥ 0	1									
3, 2	-3	≥ 0	0									

В табл. 7 значения состояний нейронов первого слоя $Sn_{i,j}^{(1)}$ определялось по выражению (3). Таблицы весов для первого слоя вычислялись на основе выражений (13), (14) и приведены на рис.36.

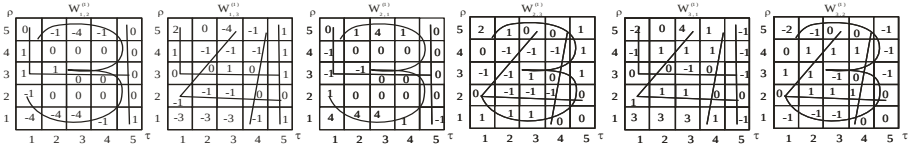


Рис. 36 Таблицы весов для первого слоя нейронной сети.

В параграфе 4.7. приводится пример сравнительного анализа на базе MNIST результатов обучения для классического многослойного персептрона и персептрона реализованного на основе метода ближайшего соседа. База MNİST включает 60000 цифр обучающей входной и выходной выборки и 10000 тестируемых входной и выходной выборки символов цифр, соответствующим образам цифр от 0 до 9. Пример реализовывался в программной среде *Python*, с использованием, встроенным в данную среду модуля *Keras*. Для создания нейронной сети случайным образом выбиралось из базы MNIST 30 эталонов (по 3 эталона от каждого образа цифры). Структура нейронной сети определялась по схеме, приведенном на рис. 33,

с дальнейшим преобразованием в полно связный многослойный персептрон. В результате количество нейронов первого слоя составило 870 нейронов, второго слоя 30 нейронов, третьего слоя – 10 нейронов. В качестве алгоритма обучения использовался стохастический алгоритм градиентного спуска. Для оценки направления градиента спуска использовалось 200 символов. Сравнительный анализ проводился на основе двух экспериментов. Полученная нейронная сеть сначала обучалось классическим путем, выбранным алгоритмом обучения – без вычисления весовых и пороговых значений. В результате этого обучения, приведенного ниже, нейронная сеть на 100-ой эпохе достигла результата 96,9% на обучающей выборке и 95% на тестируемой выборке:

Train on 54000 samples, validate on 6000 samples Epoch 1/10054000/54000
[==============================] - 9s - loss: 1,2561- acc: 0,7101 - val_loss: 1,3643 - val_acc: 0,6712 Epoch 1/10054000/54000
[==============================] - 9s - loss: 0,7112- acc: 0,8392 - val_loss: 0,7344 - val_acc: 0,8235 Epoch 2/10054000/54000
[==============================] - 11s - loss: 0,5547- acc: 0,8792 - val_loss: 0,5763 - val_acc: 0,8689 Epoch 3/10054000/54000
……
……
[==============================] - 12s - loss: 0,1121- acc: 0,9803 - val_loss: 0,1181 - val_acc: 0,9686 Epoch 99/10054000/54000
[==============================] - 12s - loss: 0,1175- acc: 0,9822 - val_loss: 0,1175- val_acc: 0,9691 Epoch 100/10054000/54000
[==============================] - 11s - loss: 0,1191- acc: 0,9728 - val_loss: 0,1209- val_acc: 0,9502 10000/10000

Для второго эксперимента использовалась та же структура нейронной сети. При этом сначала вычислялись значения весов первого слоя. Весовые и пороговые значения масштабировались в диапазон от [-1, 1]. Весовые значения второго и третьего слоя устанавливались в значения 0 или 1 по схеме, приведенном на рис. 33 и показаны на рис. 37.

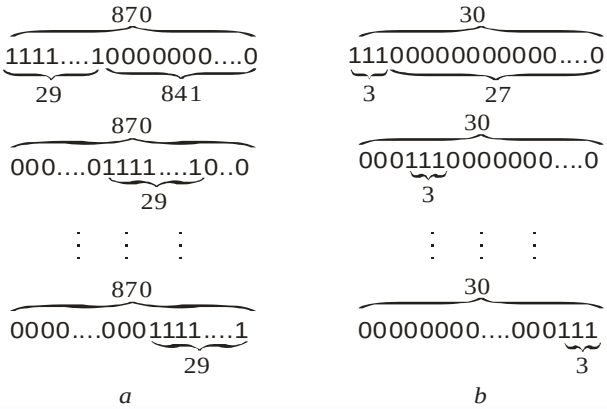


Рис. 37 (a) Значение весов для второго слоя, (b) Значение весов для третьего слоя.

Затем схема дообучалось, стохастическим алгоритмом градиентного спуска. Результаты обучения приведены в листинге:

[==============================] - 9s - loss: 0.2293 - acc: 0.9679 - val_loss: 0.1094 - val_acc: 0.9604Epoch1/2054000/54000
[==============================] - 9s - loss: 0.0814 - acc: 0.9753 - val_loss: 0.0918 - val_acc: 0.9711 Epoch 2/2054000/54000
[==============================] - 11s - loss: 0.0534 - acc: 0.9867 - val_loss: 0.0773 - val_acc: 0.9775 Epoch 3/2054000/54000
[==============================] - 12s - loss: 0.0356 - acc: 0.9879 - val_loss: 0.0622 - val_acc: 0.9807 Epoch 4/2054000/54000

[==============================] - 12s - loss: 0.0243 - acc: 0.9912 - val_loss: 0.0739 - val_acc: 0.9779 Epoch 5/2054000/54000
[==============================] - 12s - loss: 0.0221 - acc: 0.9941 - val_loss: 0.0753 - val_acc: 0.9815 Epoch 6/2054000/54000
[==============================] - 12s - loss: 0.0212 - acc: 0.9926 - val_loss: 0.0845 - val_acc: 0.9774 Epoch 7/2054000/54000
[==============================] - 13s - loss: 0.0174 - acc: 0.9915 - val_loss: 0.0884 - val_acc: 0.9772 Epoch 8/2054000/54000
[==============================] - 14s - loss: 0.0128 - acc: 0.9948 - val_loss: 0.0833 - val_acc: 0.9756 Epoch 9/2054000/54000
[==============================] - 13s - loss: 0.0118 - acc: 0.9977 - val_loss: 0.0834 - val_acc: 0.9820 Epoch 10/2054000/54000
[==============================] - 13s - loss: 0.0121 - acc: 0.9965 - val_loss: 0.0863 - val_acc: 0.9795 Epoch 11/2054000/54000
[==============================] - 13s - loss: 0.0169 - acc: 0.9954 - val_loss: 0.0781 - val_acc: 0.9836 Epoch 12/2054000/54000
[==============================] - 12s - loss: 0.0091 - acc: 0.9966 - val_loss: 0.1015 - val_acc: 0.9811 Epoch 13/2054000/54000
[==============================] - 12s - loss: 0.0071 - acc: 0.9979 - val_loss: 0.0920 - val_acc: 0.9814 Epoch 14/2054000/54000
[==============================] - 12s - loss: 0.0111 - acc: 0.9969 - val_loss: 0.1177 - val_acc: 0.9773 Epoch 15/2054000/54000
[==============================] - 12s - loss: 0.0115 - acc: 0.9967 - val_loss: 0.0831 - val_acc: 0.9808 Epoch 16/2054000/54000

[==============================] - 12s - loss: 0.0073 - acc: 0.9976 - val_loss: 0.0807 - val_acc: 0.9835 Epoch 17/2054000/54000
[==============================] - 13s - loss: 0.0089 - acc: 0.9972 - val_loss: 0.0924 - val_acc: 0.9825 Epoch 18/2054000/54000
[==============================] - 12s - loss: 0.0073 - acc: 0.9975 - val_loss: 0.0804 - val_acc: 0.9836 Epoch 19/2054000/54000
[==============================] - 12s - loss: 0.0079 - acc: 0.9969 - val_loss: 0.0815 - val_acc: 0.9831 Epoch 19/2054000/54000
[==============================] - 11s - loss: 0.0055 - acc: 0.9975 - val_loss: 0.1212 - val_acc: 0.980210000/10000

Из выше приведенного листинга, можно видеть, что на 20-ой эпохе достигается результат 98%. Этот результат выше на 2% результата предыдущего эксперимента и кроме того результат был получен в 5 раз быстрее предыдущего эксперимента (100/20=5). Кроме того, по результатам приведенного выше листинга обучения можно видеть, что уже после первой эпохи обучения результат на проверочных данных достигает значения 96%.

В параграфе 4.8. приводятся выводы касательно 4-ой главы.

В главе V рассматриваются способы и подходы применению предлагаемых архитектур к различным задачам с различными объектами распознавания, такими как символы, фотография, тексты и т.д. Рассматриваются способы представления и обработки объектов перед подачей их на входы предлагаемых нейронных сетей [1-7, 11, 16, 17, 19, 20].

В параграфе 5.1. рассматривается алгоритм выделения объектов на изображениях, фотографиях. Данный подход может позволить применение нейронных сетей к распознаванию объектов на фотографиях, сценах [7, 11].

В параграфах 5.2., 5.3., 5.4., 5.5., 5.6. рассматриваются алгоритмы выделения, обработки и представления изображений символов (Рис. 38) и текстов перед подачей выделенных объектов в нейронную сеть [1-6].

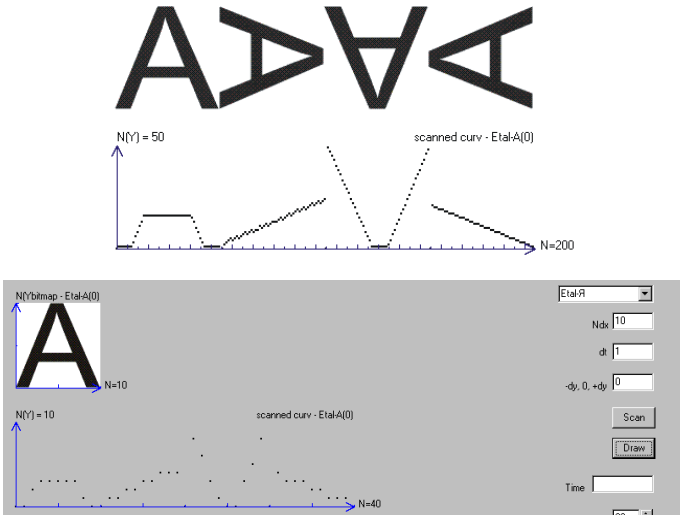


Рис. 38 Примеры кривых преобразования для печатного символа «А» с разным шагом дискретизации.

В параграфе 5.7. рассматриваются способы и возможности представления символов с наклонами, сдвигами, поворотами, которые могут быть использованы для предлагаемых нейронных сетей [19, 20].

В параграфе 5.8. рассматриваются подходы решения задачи распознавания лиц нейронными сетями на основе метрических методов распознавания. Приводятся также способ представления изображений лиц для представления на входы нейронной сети [19].

В параграфе 5.9. рассматривается возможность создания трехмерной таблицы весов, которая может быть использована, например, для представления полутонных или цветных изображений, необходимое например, в задаче распознавания фотографий [19].

В параграфах 5.10., 5.11, 5.12 рассматриваются способы и подходы применения нейронной сети на основе метрического метода распознавания для задачи анализов содержаний текстов [10, 12, 13, 16, 17, 24]. В частности, рассматривается пример анализа текстов статей на предмет определения тематики статей с целью выделения членов совета для защит диссертаций на электронном научном семинаре [10, 13, 16, 24]. Рассматривается способ представления статей на основе ключевых слов пригодных для подачи на входы нейронной сети. Нейронная сеть выполняет выбор авторов нужной тематики для научного совета. В качестве примера анализа текстов, посредством предлагаемых архитектур нейронных сетей, в параграфе рассмотрен также пример определения адресата гос. организации по содержанию текста письма заявления необходимое для задачи автоматизации процедуры обращения граждан в гос. учреждения [12], которое позволяет автоматизировать операторскую работу отбора и сортировки электронных писем. Нейронная сеть для данной задачи выбирает наименование и адресат нужной гос. организации по содержанию письма.

В главе VI на основе архитектур нейронных сетей, реализующих метрические методы распознавания, предлагается архитектура нейронной сети с попарно последовательным разделением образов [22, 23], в которой нейронная сеть разделяется на блоки, каждая из которых, по сути, выполняет работу разделения – распознавания двух образов, что значительно облегчает процедуру расширения нейронной сети и добавления в задачу распознавания новых образов без изменений предыдущих весовых и пороговых значений нейронной сети.

С этой целью в параграфе 6.1. приводится постановка данной задачи.

В параграфе 6.2. приводится архитектура двухслойной нейронной сети с попарно последовательным разделением образов, рис. 39 [22]. При этом первый слой нейронной сети на схеме рис. 39 представляет из себя отдельные блоки нейронных

сетей NN_{ij} каждый из которых состоит из двух или трех слоев нейронных сетей, (Рис. 40) и каждый блок выполняет разделение двух образов. Объедение всех результатов выходов блоков NN_{ij} происходит во втором слое, рис. 39*b*. На рис. 39*a* также показана схема подачи входных данных обучающего набора в нейронную сеть на примере задачи распознавания цифр.

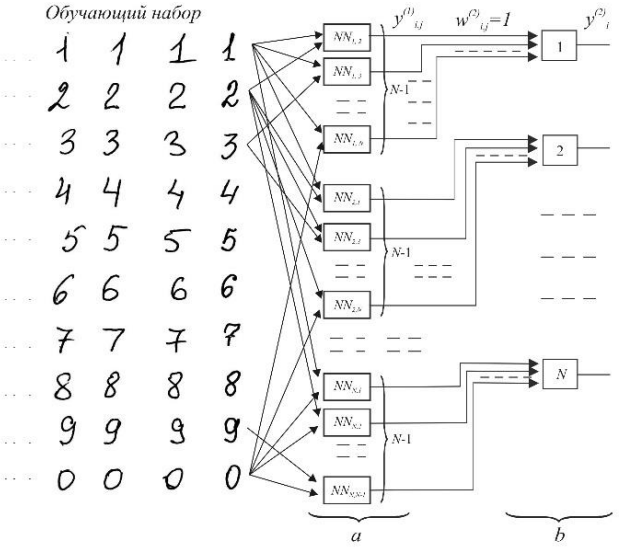


Рис. 39 Схема обучения нейронной сети.

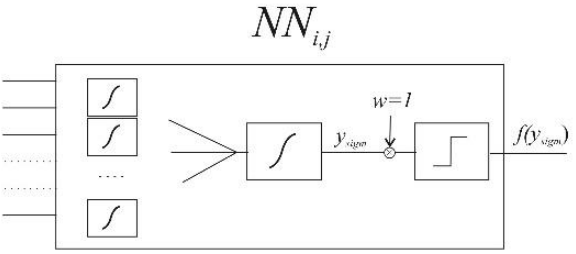


Рис. 40 Преобразование выхода $NN_{i,j}$ в бинарную форму.

Каждый блок нейронной сети может включать в себя непрерывные и пороговые нейроны, рис. 40. При этом непрерывный аналоговый выход блока нейрона преобразуется в дискретное значение {0, 1} при помощи выходного нейрона с пороговой функцией активации, рис. 40, по выражению:

$$f(y_{sigm}) = 1, if\ y_{sigm} > 0{,}5$$
$$f(y_{sigm}) = 0, if\ y_{sigm} \leq 0{,}5 \quad (56)$$

В параграфе 6.3. приводится схема сжатия полученной нейронной сети с целью уменьшения полученных блоков нейронной сети. Путем удаления блоков *NNij*, выполняющих разделение одноименных образов *{i, j}, {j, i}* количество блоков сокращается в два раза, рис. 41 [22].

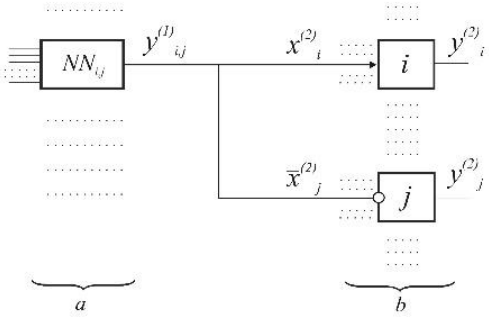


Рис. 41 Сжатие сети путем исключения блока NN_{ji}, (*a*) – первый слой, (*b*) – второй слой.

В параграфе 6.4. приводится пример создания нейронной сети с попарно последовательным разделением образов на примере простейшей задачи распознавания трех образов {A, B, C}. А также приводится формулы вычисления ошибки работы сети для данных предлагаемых сетей [23].

В параграфе 6.5. приводятся выводы по VI-ой главе.

В главе VII выполняется сравнительный анализ результатов обучений нейронной сети с вычисленными весовыми значениями и с генерацией весовых значений

случайным образом на базе MNIST. С этой целью выполняются два эксперимента, результаты сравниваются [27, 28].

В параграфе 7.1. поясняется поставленная задача.

В параграфе 7.2. создается нейронная сеть на основе метрических методов распознавания. Для этого было реализовано программный модуль в среде Builder C++ (Рис. 42), позволяющей создать и протестировать нейронную сеть метрических методов распознавания по выделенному набору эталонов [28].

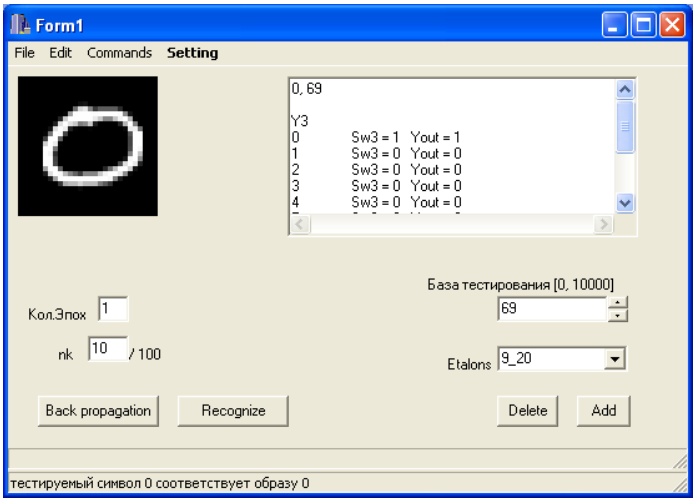


Рис. 42 Программный модуль для распознавания и обучения изображений базы MNIST, реализованный в среде Builder C++.

В качестве эталонов был использован выбранный набор из 30 образцов цифр в контрольные выборки базы MNIST (Рис. 43).

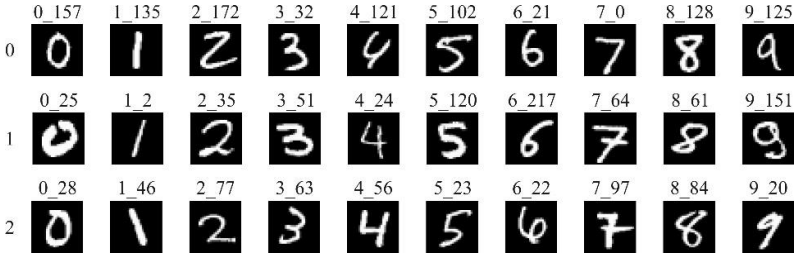
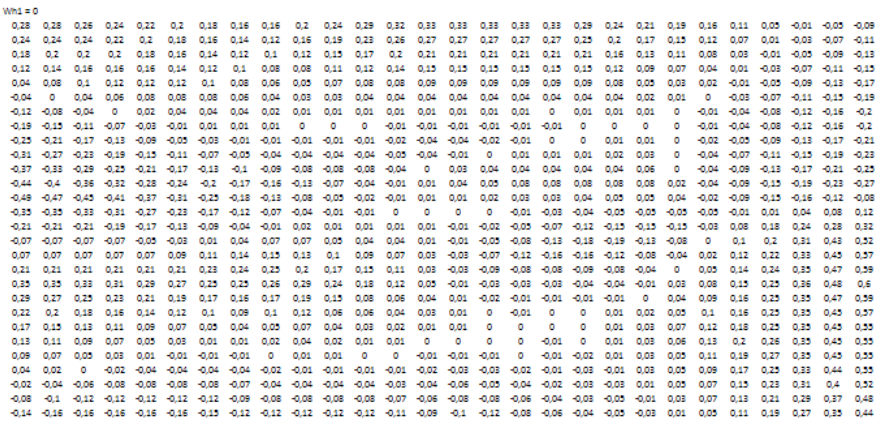
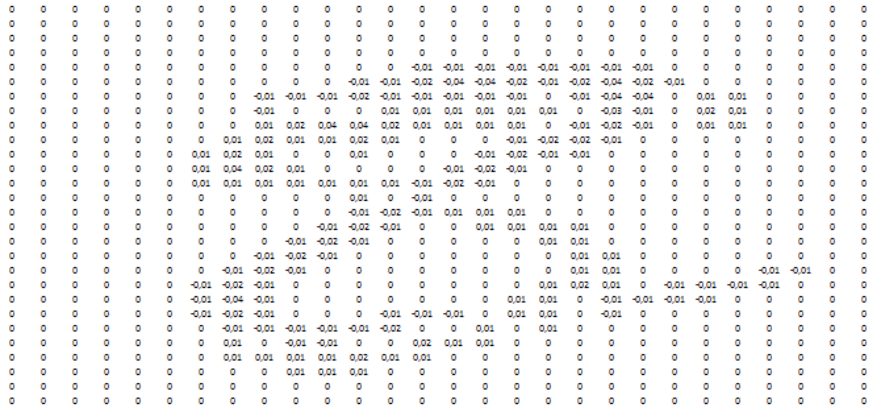


Рис. 43 Выбранные эталоны из контрольной базы MNIST, наименование изображения включает название образа и порядковый номер изображения в базе MNIST.

В данном параграфе также описано процедура вычисления значений весов для нейронов первого, второго и третьего слоя. Размерность таблиц весов нейрона первого слоя определяется размерностью изображений MNIST, пример вычисленной таблицы весов приведен на рис. 44. На рис. 45*ab* приводится также значения весов и порогов связей нейронов второго и третьего слоя.

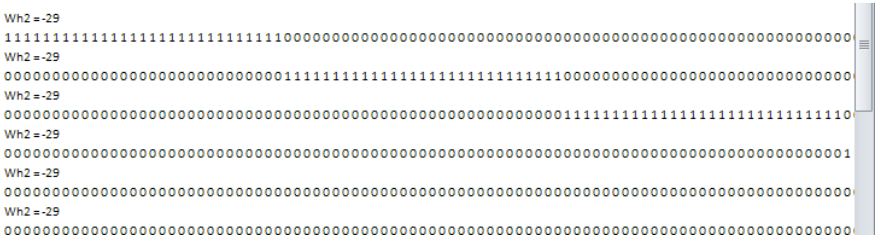


a

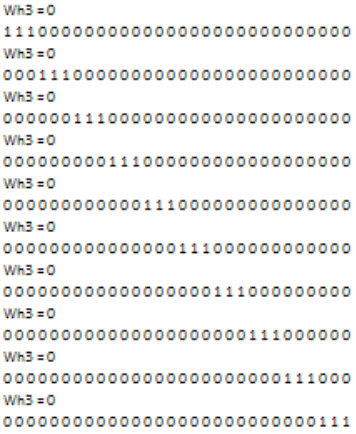


b

Рис. 44 Таблица весов первого слоя для нейрона, выполняющего сравнение эталонов 2_172 и 5_102, (*a*) для части бинарной матрицы, в которой светлым пикселям изображения (>150) соответствует 1, (*b*) для части бинарной матрицы, в которой затемненным пикселям изображения (<150) соответствует 1.



a



b

Рис. 45 (*a*) Фрагмент весов для второго слоя, (*b*) Все веса третьего слоя.

Также в данном параграфе описаны поправки и изменения, сделанные для перевода нейронов сети с пороговой функцией активации в непрерывную функцию и в частности для использования в сети сигмоидной функции активации [28]:

$$f(Sw) = \frac{1}{1+e^{-Sw}} \qquad (57)$$

В параграфе 7.3. приводятся результаты распознавания контрольной базы MNIST (10000 изображений) в полученной нейронной сети без выполнения обучения нейронной сети [27, 28]. В табл. 8 приводятся общие итоги и отдельно результаты для каждой распознаваемой цифры контрольной базы MNIST. Все вычисления, описанные в данной работе, проводились на

одном компьютере в программном модуле, приведенном на рис. 42, реализованном в среде Builder C++. На весь процесс создания нейронной сети и вычисления всех весов в программном модуле на рис. 42 было зафиксировано общее потраченное время $t_{созд}$ = 0,5469 с., то есть меньше секунды.

Табл. 8 Результаты распознавания контрольной базы MNIST (10000 символов) без обучения

s0 = 834	i0 = 980	p0 = 85%
s1 = 968	i1 = 1135	p1 = 85%
s2 = 530	i2 = 1032	p2 = 51%
s3 = 454	i3 = 1010	p3 = 44%
s4 = 410	i4 = 982	p4 = 41%
s5 = 411	i5 = 892	p5 = 46%
s6 = 586	i6 = 958	p6 = 61%
s7 = 556	i7 = 1028	p7 = 54%
s8 = 773	i8 = 974	p8 = 79%
s9 = 750	i9 = 1009	p9 = 74%
Итого: i = 10000, s = 6272, p = 62%		

В параграфе 7.4. приводится описание стохастического алгоритма обучения - обратного распространения ошибки (back propagation), реализованного в программном модуле на рис. 42. Показаны также особенности обращения к нейронам для схемы на рис. 33 при реализации алгоритма back propagation для данной сети [28].

В параграфе 7.5. приведены результаты сравнения двух выполненных эксперимента полученной нейронной сети, реализованного в программном модуле на рис. 42 [27, 28]. В первом эксперименте выполнялось обучение полученной нейронной сети с уже вычисленными весовыми и пороговыми значениями. Во втором эксперименте на основе той же нейронной сети выполнялось обучение нейронной сети классическим образом – то есть без использования предварительного вычисления значений весов и порогов. Для второго эксперимента значения весов определялись случайным образом. Результаты сравнения приведены в табл. 9 - для

обучающей базы MNIST – 60000 изображений, и в табл. 10 - для контрольной базы MNIST – 10000 изображений цифр. По результатам таблиц можно видеть, что для случая обучения нейронной сети с предварительно вычисленными значениями весов результаты на всех эпохах обучения лучше, чем для классического способа обучения.

Табл. 9 Сравнение результатов обучения нейронной сети по обучающей выборке MNIST (60000 изображений) для каждой эпохи обучения.

		обучение нейронной сети с предварительно вычисленными весами				обучение нейронной сети с случайной инициализацией весов в диапазоне [-0,5; 0,5]			
№ Эпохи	Скорость обучения	Кол. узн. изоб.	% узн. изоб.	S_{err}	время, мин.	Кол. узн. изоб.	% узн. изоб.	S_{err}	время, мин.
1	0,1	43932	73%	1199	159	35370	59%	1935	256
2	0,1	49748	83%	737	98	46033	76%	1051	139
3	0,02	52285	87%	545	72	49195	82%	784	104
Общее время обучения, в минутах					329	Общее время обучения, в минутах			499

Табл. 10 Сравнение результатов обучения нейронной сети с проверкой на контрольной выборке MNIST (10000 изображений) для каждой эпохи обучения.

№ Эпохи	Скорость обучения в эпохе	Обучение нейронной сети с предварительно вычисленными весами	Обучение нейронной сети с случайной инициализацией весов
1	0,1	9145	8894
2	0,1	9282	9116
3	0,02	9449	9256

Кроме того, время, потраченное на обучение нейронной сети с вычисленными весовыми значениями, также меньше на всех эпохах по сравнению с нейронной сетью, обучаемой с нуля, рис. 46. В целом на обучение нейронной сети с вычисленными весовыми значениями было затрачено 329 мин = 5 час.29 мин., при обучении нейронной сети с нуля общее время обучения

составило 499 мин. = 8 час.19 мин. То есть для второго эксперимента понадобилось почти на 3 часа больше времени. Эти полученные данные подтверждают процесс ускорения процедуры создания и обучения нейронной сети с предварительно вычисленными значениями весов и порогов нейронной сети.

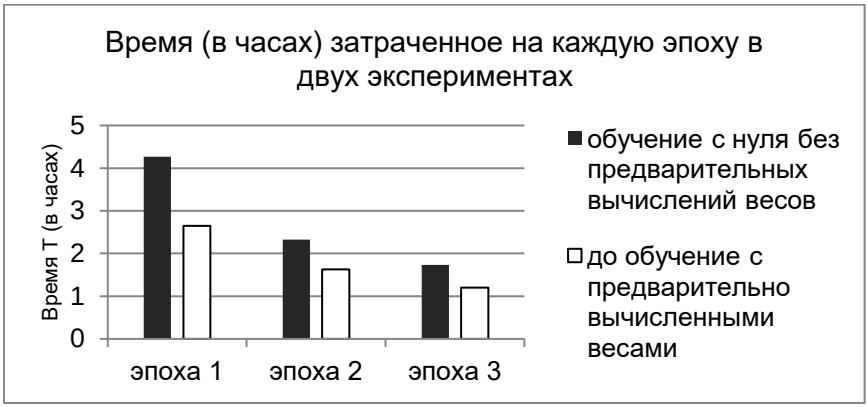


Рис. 46 Диаграмма сравнения затраченного времени на каждую эпоху для двух экспериментов.

В параграфе 7.6 приводятся выводы полученных экспериментов.

В главе VIII рассматривается возможность практически мгновенного определения значений весов нейронной сети (без вычисления и обучения) на основе параметров электростатического поля [29, 30].

В параграфе 8.1. описана поставленная цель решаемая в 8-ой главе.

В параграфе 8.2. предлагается использовать в качестве меры близости формулу электростатики – напряженность [29]. Показаны, как на основе формулы напряженности могут вычисляться значения весов (58). Рассмотрены проблемные стороны такого подхода и способы их разрешения.

$$w_{i,j}^{(1)} = \overline{E}_1 - \overline{E}_2 = \frac{K \otimes q_1}{d_1^2} - \frac{K \otimes q_2}{d_2^2}, \qquad (58)$$

При этом значения напряженности электростатического поля определяется датчиками напряженности, значения с которых поступают на входы нейронной сети (Рис. 47).

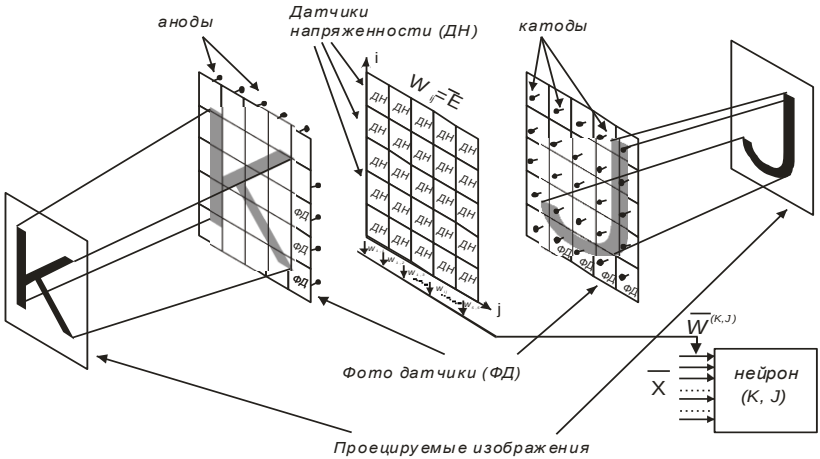


Рис. 47 Схема, демонстрирующая получение весовых значений нейрона первого слоя для проецируемых изображений *«K, J»* при помощи фотодатчиков и датчиков напряженности.

В параграфе 8.3. предлагается также использовать в качестве меры близости формулу электростатики – потенциал [29]. Показано как на основе формулы потенциала могут вычисляться значение весов (59), рис. 48, 49.

$$\varphi_{(i,j)}^{Sum} = \sum_{\substack{i=0,\\ j=0}}^{\substack{i=28,\\ j=28}} \varphi_{(i,j)}^{(Img1)} - \sum_{\substack{i=0,\\ j=0}}^{\substack{i=28,\\ j=28}} \varphi_{(i,j)}^{(Img2)} = \sum_{\substack{i=0,\\ j=0}}^{\substack{i=28,\\ j=28}} \frac{K*q_{(i,j)}^{(Img1)}}{r_{(i,j)}^{(Img1)}} -$$

$$- \sum_{\substack{i=0,\\ j=0}}^{\substack{i=28,\\ j=28}} \frac{K*q_{(i,j)}^{(Img2)}}{r_{(i,j)}^{(Img1)}}, \qquad (59)$$

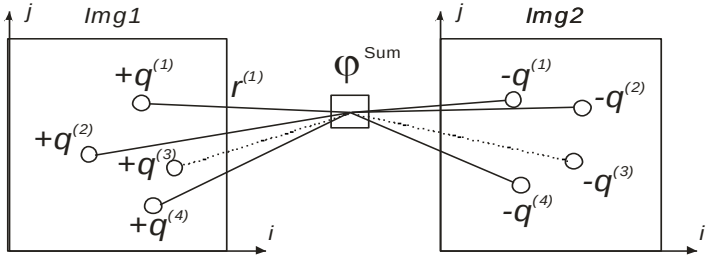


Рис. 48 Схема определения потенциала-веса в одной ячейке таблицы весов (плоскости датчиков потенциометров) первого слоя.

При этом значения потенциала электростатического поля определяются датчиками потенциометрами, значения с которых поступают на входы нейронной сети (Рис. 49).

В параграфе 8.4. выполняются эксперименты по определению значений весов нейронной сети на основе имитированного электростатического поля [30]. Эксперименты были реализованы в программной среде Builder C++.

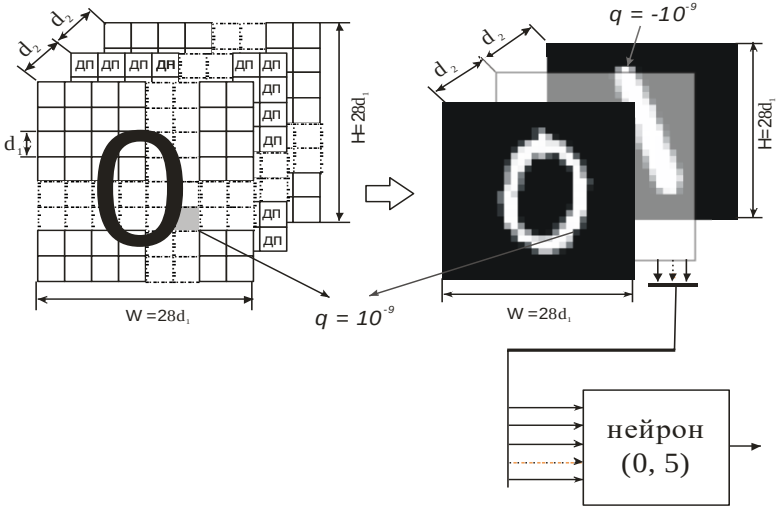


Рис. 49 Схема имитируемой модели заряженных плоскостей изображений 0_157 и 1_46 из базы MNIST и плоскости датчиков потенциометров (дп), определяющий нейрон первого слоя нейронной сети.

Табл. 11 Таблицы весов нейрона первого слоя, выполняющего сравнение изображений 0_157, 1_46 базы MNIST. Значения весов нейронной сети определены суммарным потенциалом.

k = 0, k1 = 5, 0_157, 1_46 Wh1 = -26241															
1	2	3	4	5	6	7	8	9	10	11	12	13	…..	27	28
218	217	216	204	192	182	160	139	113	98	101	129	188		395	378
227	226	221	212	195	175	146	105	58	16	-2	33	122		424	401
241	240	232	221	202	171	130	69	-19	-116	-188	-162	-14		456	427
256	255	248	237	215	178	124	36	-108	-316	-529	-544	-284		486	447
274	275	274	259	238	199	139	28	-172	-530	-986	-1063	-638		519	474
295	300	297	291	275	243	184	70	-145	-556	-1095	-1169	-626		549	501
318	327	333	331	323	305	263	180	8	-319	-754	-792	-267		578	530
342	358	373	376	386	389	375	343	272	146	-26	-157	-18		610	551
372	390	412	438	461	487	513	542	582	655	673	283	-81		639	573
396	426	458	498	538	592	663	758	889	1072	1119	492	-350		663	591
429	464	503	558	616	699	818	978	1209	1469	1436	597	-711		680	605
456	499	549	615	695	803	963	1192	1539	1911	1668	545	-893		693	610
480	528	592	665	763	898	1088	1368	1789	2201	1857	642	-782		693	609
504	557	625	709	823	977	1187	1500	1959	2380	2018	793	-646		683	604
519	579	653	745	868	1039	1264	1601	2079	2519	2166	953	-484		666	590
534	597	675	770	896	1075	1319	1666	2166	2632	2312	1148	-228		639	566
543	609	687	787	916	1095	1345	1702	2223	2721	2457	1382	151		605	541
545	610	692	791	916	1092	1345	1701	2229	2769	2580	1627	553		563	506
544	611	685	781	902	1070	1309	1650	2164	2734	2648	1873	991		521	472
536	600	673	761	878	1030	1238	1544	1986	2505	2584	2156	1538		474	440
527	584	656	734	836	970	1148	1385	1700	2052	2311	2440	2182		430	409
514	568	628	698	791	901	1042	1221	1439	1693	2008	2370	2441		391	382
498	544	600	663	741	832	942	1076	1233	1417	1655	1919	2125		359	353
481	521	570	625	686	763	850	951	1063	1197	1347	1508	1676		334	328
463	496	537	589	639	695	767	841	924	1014	1109	1211	1300		315	314
439	472	505	549	592	639	691	748	805	866	927	985	1025		300	296
419	447	480	512	545	590	625	669	705	748	788	817	832		289	286
398	423	449	477	508	541	568	598	627	654	679	694	695		278	275

В параграфе 8.5. выполняется проверка результатов экспериментов [30]. Результаты показали, что при использовании в качестве значений весов потенциалов электростатического поля (табл. 11), количество правильно распознанных элементов контрольной базы MNIST составляет более чем 50% на 30-и эталонах и увеличивается с увеличением количества используемых эталонов (табл. 12), а также изменяется при изменение физических показателей потенциала электростатического поля, таких как значения точечных зарядов и расстояний между панелями проектируемых изображений.

Табл. 12 Результаты тестирования на контрольной базе MNIST с использованием 50 эталонов для схемы на рис. 49 где точечный заряд $q = 10^{-9}$ Кл. , расстояние между точечными зарядами и заряженными панелями равно $d_2 = 4$ см., $d_1 = 2$ см.

s0 = 852	i0 = 980	p0 = 87%
s1 = 1123	i1 = 1135	p1 = 99%
s2 = 516	i2 = 1032	p2 = 50%
s3 = 585	i3 = 1010	p3 = 58%
s4 = 775	i4 = 982	p4 = 79%
s5 = 588	i5 = 892	p5 = 66%
s6 = 555	i6 = 958	p6 = 58%
s7 = 616	i7 = 1028	p7 = 60%
s8 = 633	i8 = 974	p8 = 65%
s9 = 726	i9 = 1009	p9 = 72%
10000, 6969, 70%		

В параграфе 8.6. приводятся оценка результатов данной главы.

В Итоговых выводах диссертационной работы приводятся, основные результаты и заключения, основанные на проведенных исследованиях диссертационной работы. Приводится научная новизна и практическая ценность работы.

В Приложении диссертационной работы приведены, разработанные программные коды в среде Builder C++, в которых реализуется нейронную сеть на основе метрических методов распознавания, вычисляются все значения весов, реализуется определение весов на основе параметров электростаического поля и в которых выполнялись, приведенные в работе сравнительные эксперименты и оценки. Кроме того в приложении приводятся таблицы весов нулевого и первого слоя, а также значения всех выходов (функций состояний и активаций нейронной) реализованной нейронной сети для отдельно выбранных входных распознаваемых элементов нейронной сети из базы MNIST.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Предложены новые архитектуры нейронных сетей прямого распространения с вычисляемыми параметрами.
2. Приведены математические модели предложенных архитектур нейронных сетей.
3. Предложены способы минимизации количества нейронов и связей предлагаемых нейронных сетей.
4. Предложены аналитические выражения вычисляющие параметры структуры нейронной сети: количество нейронов, слоев, связей.
5. Предложены аналитические выражения, позволяющие вычислять значения весовых и пороговых значений связей нейронов.
6. Разработаны схемы и алгоритмы применения предлагаемых нейронных сетей в многозадачных применениях.
7. Разработаны алгоритмы и схемы применения предлагаемых нейронных сетей в задачах с нечеткими выводами.
8. Найдены и приведены условия, определяющие распределение весовых и пороговых значений многослойного персептрона.
9. На основе алгоритма обучения Видроу-Хоффа показано и

математически обосновано стремление алгоритмов обучения нейронных сетей прямого распространения к условиям распределения весовых значений, полученных для предлагаемых вычисляемых нейронных сетей.

10. Предложен общий алгоритм вычисления весовых и пороговых значений нейронов второго и третьего слоя многослойного персептрона на основе метрических методов распознавания.
11. На основе архитектур нейронных сетей метрических методов распознавания предложены архитектуры нейронных сетей реализующие попарно последовательное разделение образов.
12. Реализован программный модуль в программной среде Builder C++, реализующий нейронные сети на основе метрических методов распознавания.
13. На базе цифр MNIST показано, что за доли секунды создается нейронная сеть и вычисляются все весовые значения таблиц весов.
14. Было показано, что для нейронов данных архитектур применимы разные функции активации, в том числе и непрерывные функции активации, такие как сигмоидные функции активации.
15. Практически также было показана, что данная нейронная сеть может также дообучаться алгоритмом обратного распространения ошибки back propagation. Для этого был реализован и показан стохастический алгоритм - back propagation, применительно к предложенной архитектуре нейронной сети.
16. На базе MNIST экспериментально доказано ускорение процедуры создания и обучения нейронной сети с предварительно вычисленными значениями весов по сравнению с классическим подходом.
17. Показана теоретически и экспериментально возможность практически мгновенного определения значений весов без обучения и вычисления весовых значений, на основе таких параметров электростатического поля, как напряженность и потенциал.

Основное содержание диссертационной работы изложено в следующих публикациях:

1. Гейдаров, П.Ш. Структурно признаковый метод распознавания рукопечатных символов // Известия НАН Азерб., серия физико математических и технических наук, — 2006. Т. XXV, №3, — с. 65-70.
2. Нусратов, О.К., Гейдаров, П.Ш. Позиционно-бинарный метод распознавания рукопечатных символов // Труды VII Международного симпозиума «Интеллектуальные системы», INTELS-2006, — Краснодар, Россия, — 2006, — с. 474-477.
3. Нусратов, О.К, Гейдаров, П.Ш. Метод распознавания рукопечатных символов и текстов // Труды Международной Конференции PCI, 2006 «Проблемы Кибернетики и информатики», — Баку: — 2006. — с. 52-58.
4. Нусратов, О.К. Метод распознавания рукопечатных символов, Решение государственного агенства по стандартизации, метрологии и патента Азербайджанской Республики,02.10.2006 A 2006 0182 / Гейдаров П.Ш. — 2006.
5. Гейдаров, П.Ш. Алгоритмы устранения ошибочных сегментаций в структурно-сегментном представлении символа // Известия НАН Азерб., серия физико математических и технических наук, — 2008. Т. XXVIII, №3, — с. 115-119.
6. Nusratov, O.Q. Position-Width-Pulse Algorithm for Recognition of Handwritten Characters / P.Sh. Geidarov // Automatic Control And Computer Sciences, — 2007. V. 41, №6, — pp.332-336. **(Springer, Scopus, Web of Science: ESCI)**
7. Geidarov, P.Sh. Algorihm determination of spacing interval up to object // The second international conference “Problem of Cybernetics and informaties.” — V. 2, 2008. — p. 68-71.
8. Geidarov, P.Sh. Neural networks on the basis of the sample method // Automatic Control And Computer Sciences, — 2009. V. 43, № 4, — p. 203-210. **(Springer, Scopus, Web of Science: ESCI)**

9. Гейдаров, П. Ш. Нейронные сети на основе метрических методов распознавания в применении к задачам с нечеткими выводами // Искусственный интеллект и принятия решений, — 2010. № 2, — с. 77-88. **(Web of Science : RSCI)**

10. Гейдаров, П.Ш. О возможностях электронного научного семинара // Социологические исследования, — 2010. № 8, — с. 135–137. **(Web of Science : SSCI)**

11. Гейдаров, П.Ш. Алгоритм определения расположения и размеров объекта на основе анализа изображений объектов // Компьютерная оптика, — Саратов: — 2011. Т. 35, №2, — с. 275-280. **(Scopus, Web of Science: ESCI)**

12. Гейдаров, П.Ш. Автоматизация процедуры обращений граждан в госучреждения и электронное правительство // Социологические исследования, — 2012. № 11. — с. 71-80. **(Web of Science : SSCI)**

13. Гейдаров, П.Ш. Электронный научный семинар // Elektron elm problemləri üzrə 1 respublika elmi-praktiki konfransın materialları, — Bakı: — 2012. — с. 54-57.

14. Geidarov, P.Sh. Multitasking application of neural networks implementing metric methods of recognition // Autom. Remote Control, — 2013. V. 74, № 9, — p. 1474–1485. (**Scopus, Web of Science: SCI Exp.)**

15. Geidarov, P.Sh. Clearly Defined Neural Networks Architecture // Optical Memory & Neural Networks, — 2015. V. 24, I. 3, — p. 209-219. **(Springer, Scopus)**

16. Гейдаров, П. Ш. Балловая система оценки научных трудов и электронный научный семинар // Социологические исследования, — 2015. № 4. — с. 162-167. **(Web of Science : SSCI)**

17. Гейдаров П.Ш. Алгоритм кратчайшего маршрута на основе выделенного набора маршрутов // Информационные технологии, — 2016. Т. 22, №9, — с. 660–668. **(Web of Science : RSCI)**

18. Geidarov, P. Sh. Clearly defined architectures of neural networks and multilayer perceptron // Optical Memory and Neural Networks, — 2017. V. 26, — p. 62–76. **(Springer, Scopus)**

19. Гейдаров, П.Ш. Алгоритм реализации метода ближайшего соседа в многослойном персептроне // Труды СПИИРАН, — 2017. Т. 51, — с. 123-151. **(Scopus)**

20. Гейдаров, П.Ш. Перспективы создания государственной системы мобильного видеомониторинга // Информационное общество, — 2017. №4-5. — с. 114-121. **(Web of Science : RSCI)**

21. Гейдаров, П.Ш. Нейронные сети прямого распространения с вычисляемыми параметрами // Информационные технологии, — 2017. Т. 23, № 7, — с. 543-552. **(Web of Science : RSCI)**

22. Гейдаров, П.Ш. Архитектура нейронной сети с попарно последовательным разделением образов. // Прикладная дискретная математика, — 2018. №41, — с. 98-109. **(Scopus, Web of Science : ESCI)**

23. Geidarov, P.Sh. Neural Networks with Image Recognition by Pairs // Optical Memory and Neural Networks, — 2018. V. 27, № 2, — p. 113–119. **(Springer, Scopus)**

24. Гейдаров, П.Ш. Автоматизированный выбор состава научного совета электронного научного семинара с использованием нейронных сетей на основе метрических методов распознавания // Международная научно-практическая конференция “Научно-практические исследования”, — Россия, Омск: — 2020. № 1-4, Т. 24, — с. 35-42.

25. Гейдаров, П.Ш. Нейронные сети на основе метрических методов распознавания // XXVI Международная научно-практическая конференция «Advances in Science and Technology», — Россия, Москва: — 2020. Часть I, – с. 77-81.

26. Гейдаров, П.Ш. Алгоритм вычисления значений весов синапсов первого слоя нейронной сети на основе метрических методов распознавания. Часть 1. // Информационно-Управляющие Системы, — Санкт-Петербург: — 2020. №2, — с. 20-30. **(Scopus)**

27. Гейдаров, П.Ш. Алгоритм вычисления значений весов синапсов первого слоя нейронной сети на основе метрических методов распознавания. Часть 2. // Информационно-

Управляющие Системы, — Санкт-Петербург: — 2020. №3, — с. 25-38. **(Scopus)**

28. Geidarov, P. Sh. Comparative Analysis of the Results of Training a Neural Network with Calculated Weights and with Random Generation of the Weights // Automation and Remote Control, NewYork, — 2020. V. 81, №. 7, — с. 1211-1229. **(Scopus, Web of Science : SCI Exp.)**

29. Гейдаров, П.Ш. О возможности определения значений весов нейронной сети электростатическим полем // Искусственный интеллект и принятие решений, — Москва: – 2021. №1, – с. 33-49. **(Web of Science : RSCI)**

30. Гейдаров, П.Ш. Эксперимент по созданию нейронной сети с весами, определяемыми потенциалом имитированного электростатического поля // Искусственный интеллект и принятие решений, — Москва: – 2021. №2. – с. 78-92. **(Web of Science : RSCI)**

Защита диссертации состоится 26 ноября 2021 года в 14^{00} на заседании Диссертационного совета ED 1.20 действующего на базе Института Систем Управления НАН Азербайджана

Адрес: Az 1141, ул. Б. Вагабзаде, 9

С диссертацией можно ознакомиться в библиотеке Института Систем Управления НАН Азербайджана

Электронные версии диссертации и автореферата размещены на официальном сайте Института Систем Управления НАН Азербайджана.

Автореферат разослан по соответствующим адресам 21 октября 2021 года.

Подписано в печать: 7.11.2021

Формат бумаги: А5

Объём: 89000

Тираж: 70