

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

**ИНТЕЛЛЕКТУАЛЬНЫЙ МЕТОД РАСПОЗНАВАНИЯ
ТЕКСТОВОЙ ИНФОРМАЦИИ НА УПАКОВКАХ ТОВАРОВ**

Магистерская диссертация
обучающегося по направлению подготовки
09.04.02 Информационные системы и технологии
заочной формы обучения,
группы 07001573
Беспаловой Анастасии Валерьевны

Научный руководитель
к. т. н., доцент
Шамраев А.А.

Рецензент
к.т.н., доцент
Зайцева Т.В.

БЕЛГОРОД 2018

РЕФЕРАТ

Интеллектуальные методы распознавания текстовой информации на упаковках товаров. Беспалова Анастасия Валерьевна, магистерская диссертационная работа, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 68, включая приложения 75, количество рисунков 27, количество использованных источников 41.

КЛЮЧЕВЫЕ СЛОВА: фальсифицированное лекарственное средство, искусственная нейронная сеть, сверточная нейросеть.

ОБЪЕКТ ИССЛЕДОВАНИЯ: интеллектуальный метод распознавания текста.

ПРЕДМЕТ ИССЛЕДОВАНИЯ: разработка интеллектуального метода распознавания текстовой информации на упаковках товаров.

ЦЕЛЬ РАБОТЫ: разработка интеллектуального метода распознавания даты и серии на упаковках товаров, для выявления, просроченного и фальсифицированного лекарственного средства.

ЗАДАЧИ ИССЛЕДОВАНИЯ: проанализировать существующие методы распознавания текста, исследовать архитектуры нейросети. На основе архитектуры спроектировать и разработать нейросеть, затем обучить и протестировать полученную нейросеть, а также сравнить полученные результат с данными из имеющейся

МЕТОДЫ ИССЛЕДОВАНИЯ: классические методы распознавания, сверточные сети, методы обучения.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: разработан интеллектуальный метод на основе сверточной нейросети, который позволяет распознать текст на упаковке товаров и сравнить с имеющейся базой данных.

СОДЕРЖАНИЕ

| | |
|--|----|
| ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ | 5 |
| ВВЕДЕНИЕ..... | 6 |
| 1 Анализ состояния проблем и постановка задачи исследования | 8 |
| 1.1 Анализ рынка лекарственных препаратов | 8 |
| 1.2 Анализ состояния оборота фальсифицированных средств | 10 |
| 1.3 Государственное регулирование фармацевтической отрасли..... | 13 |
| 1.4 Постановка задачи..... | 18 |
| 2 Методы и модели распознавания текстовой информации | 20 |
| 2.1 Классические методы распознавания образов | 20 |
| 2.2 Нейросетевые методы распознавания информации..... | 31 |
| 2.3 Выбор и обоснование архитектуры и алгоритма обучения искусственной нейронной сети | 38 |
| 2.3.1 Анализ архитектур сверточных нейронных сетей..... | 38 |
| 2.3.2 Выбор алгоритма сегментации изображения..... | 39 |
| 2.3.3 Выбор алгоритма обучения..... | 43 |
| 3 Программная реализация нейросетевой архитектуры | 48 |
| 3.1 Формирование сверточной архитектуры искусственной нейронной сети | 48 |
| 3.2 Алгоритмы и технология решения задач..... | 49 |
| 3.4 Процесс обучения полученной нейросети | 51 |
| 3.5 Правила пользования разработкой конечным пользователем | 52 |
| 4 Тестирование нейросетевой архитектуры | 60 |
| 4.1 Характеристики исходных данных | 60 |
| 4.3 Результаты эксперимента..... | 62 |
| ЗАКЛЮЧЕНИЕ | 64 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... | 66 |
| ПРИЛОЖЕНИЕ А | 70 |

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

GMP – надлежащая производственная практика

БД – база данных

ИНС – искусственная нейронная сеть

ЛС – лекарственное средство

СНС – сверточная нейронная сеть

ФЛС – фальсифицированное лекарственное средство.

ВВЕДЕНИЕ

Интеллектуальные методы распознавания образов представляют собой наиболее математически направленный раздел теории искусственного интеллекта, в котором решаются проблемы, связанные с классификацией объектов различных форм и оттенков. Распознавание образов одна из тех задач, которые без особого труда решаются «естественным» интеллектом. Благодаря этому многие ученые стали заниматься разработкой методов и алгоритмов автоматизированного решения этой задачи. В частном случае, когда образы легко определяются конечным набором признаков, границы классов точно описываются, а сами классы не пересекаются, степень неопределенности можно считать минимальной, и задачу классификации можно решать, не учитывая неопределенность данных. [9,11]

Одними из таких интеллектуальных методов распознавания текстовой информации является искусственные нейронные сети (ИНС). ИНС представляют собой математическую модель функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Как и в биологической нейронной сети, основным элементом искусственной нейросети является нейрон. Соединенные между собой нейроны образуют слои, количество которых может варьироваться в зависимости от сложности нейросети и решаемых ею задач. Кроме того, нейронные сети не программируются в привычном смысле этого слова, они обучаются. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных. [8,16]

Актуальность данного метода обусловлена необходимостью распознавания текста на упаковках товаров. Исходя из того, что рынок подпольных фармацевтических фабрик продолжает расти, несмотря на предпринимаемые меры по ужесточению контроля за производством и сертификацией лекарственных товаров. Поддельная продукция наносит колоссальный ущерб фармацевтическим компаниям и здоровью пациентов. К сожалению, среди поддельных лекарств нередко встречаются опасные, угрожающие жизни аналоги брендовых препаратов. Для того чтобы снизить риск поступления некачественных или просроченных фирменных товаров на прилавки аптек и аптечных киосков, рентабельно, с помощью искусственной нейронной сети распознать серию и дату на упаковке и сравнить полученные результаты с введенными ранее сведениями о препарате в базу данных.

Целью создания и использования метода нейронной сети является: сокращения времени на сортировку и дальнейшую обработку упаковок, изъять из товарооборота некачественные лекарственные препараты, выявить точные нормативные инструменты для оперативной корректировки и переформирования документооборота в связи с изменением производственных ситуациях.

1 Анализ состояния проблем и постановка задачи исследования

1.1 Анализ рынка лекарственных препаратов

Фармацевтический рынок традиционно входит в число наиболее рентабельных рынков, как в России, так и за рубежом. Обилие фальсификата и активное проникновение на российский рынок иностранных компаний - лишние тому подтверждения.

Его развитие тесно связано с общей социально-экономической ситуацией в стране, прежде всего, динамикой реальных доходов населения. Как показывает опыт наших европейских соседей, с ростом дохода возрастает потребление лекарственных средств (ЛС). Они относятся к продуктам первой необходимости. Однако по мере увеличения доходов населения изменяется структура потребления ЛС. Дешевые препараты-дженерики заменяются дорогостоящими оригинальными препаратами, преимущественно импортными, кроме того изменяется структура по группам: возрастает доля витаминов, биологически активных добавок и т.п.

В 2003 г. объем розничного рынка оценивался суммой примерно 148 млрд. руб. в ценах конечного потребления, по данным компании "ФармАналитик", за 1 квартал 2004 г. объем рынка составил 41 млрд. руб., что выше аналогичного показателя прошлого года на 17%. Львиная доля рынка в стоимостном выражении приходится на продукцию иностранных производителей. Ежегодный прирост объема рынка достигает около 10-15%.

Фармацевтический рынок включает в себя 3 сегмента: производство, дистрибьюция и розничные продажи. Вслед за мировым фармацевтическим рынком отечественный рынок имеет тенденцию к росту уровня концентрации, формированию вертикальных и горизонтальных интегрированных структур, оптовых и розничных сетей.

В секторе отечественного производства уровень концентрации пока невысок. На долю 2-х крупнейших производителей (компании ОАО "Отечественные лекарства", ICNRussia) составляет приблизительно 6% отраслевого выпуска.

Для российского фармрынка характерна относительно высокая численность оптовиков, это, по большей части, мелкие и средние предприятия. В перспективе их количество сократится, по мнению экспертов, до 10-20. В настоящее время на долю 2-х крупнейших дистрибьюторов ("Протек" и SIA International) приходится более 30% продаж. [1][2]

В розничном сегменте рынка до 40% составляют государственные и муниципальные аптеки. Однако в ближайшем будущем ситуация изменится. Доля госсобственности сократится, значительно вырастет значимость аптечных сетей, таких как "Аптеки 36,6", "Чудо-Доктор" и т.п.

На российском рынке формируются фармацевтические холдинги, например, "Отечественные лекарства" (объединяет предприятия-производителей), "Профит Хаус" (производство и розница), "Аптеки 36,6" (ЗАО "Аптеки 36,6", ЗАО "Верофарм").

Региональная структура розничного рынка ЛС отличается неравномерностью распределения, что, однако, соответствует неравномерному распределению населения по территории РФ. Более детальное изучение структуры потребления позволило выявить несколько более высокое подушевое потребление ЛС в крупных регионах, таких как Москва, Московская область, Петербург и т.п.

По данным "ROMIRMonitoring", 44% россиян предпочитает отечественные ЛС, для 42% не имеет значение страна-изготовитель. В малых городах и в сельской местности доля приверженцев отечественной фармацевтической продукции выше. По-видимому, кроме психологических факторов на выбор потребителей влияет стоимость препаратов. Российские аналоги зарубежных препаратов существенно дешевле. Действительно, даже

в Москве рыночная доля отечественных ЛС в натуральном выражении достигает 69% при доле в стоимостном выражении менее 20%.

Российский фармацевтический рынок переживает не самые спокойные времена. С одной стороны, предстоящий переход на стандарты GMP, с другой - рост конкуренции со стороны иностранных производителей. Оба фактора потребуют от российских производителей максимального напряжения сил и больших инвестиций в инновации, производство и продвижение продукции.

До последнего времени российское производство было ориентировано на выпуск препаратов-дженериков, при этом преобладали традиционные старые дженерики, доступные по цене большинству населения.

Серьезная угроза для легальной активности рынка - распространение фальсифицированных лекарственных средств (ФЛС). Примерно 12% фармакологических товаров - подделки. Доля фальсификата продолжает возрастать. Ежегодные потери производителей по этой причине оцениваются суммой порядка 250 млн. долл.

1.2 Анализ состояния оборота фальсифицированных средств

Как было изложено ранее, угроза лекарственной безопасности здоровья населения связана, прежде всего, с оборотом фальсифицированных лекарственных средств. На российском фармацевтическом рынке их количество весьма значительно.

В рамках мероприятий Росздравнадзора ежегодно из обращения изымается свыше 30 торговых наименований 80 серий ФЛС, 6 наименований 180 серий фальсифицированных фармацевтических субстанций и 1000 серий готовых лекарственных средств, произведенных из данных субстанций. Видно, что фармацевтический рынок переполнен лекарственными фальсификатами низкой себестоимости, легко конкурирующими с легальной фармацевтической продукцией. ФЛС являются опасными, прежде всего, потому, что принимаемые потребителем как жизненно необходимые средства,

они не дают терапевтического эффекта и могут привести к безуспешному лечению, создать угрозу для здоровья или даже жизни. Несмотря на то, что ущерб от оборота ФЛС предположить и подсчитать трудно, федеральные и региональные бюджеты не получают миллиарды рублей дохода и прибыли, которые оседают в «карманах» организованной преступности. [3][4]

Таким образом, оборот ФЛС наносит ощутимый вред здоровью, жизни и социальному положению населения, авторитету органов власти и управления, а также международному престижу, инвестиционному положению, экономике страны.

Сложившаяся на фармацевтическом рынке обстановка требует активизации противодействия данным преступлениям. Норма, предусматривающая уголовную ответственность непосредственно за обращение ФЛС, была включена в Уголовный кодекс Российской Федерации (далее – УК РФ) федеральным законом от 31.12.2014 № 532–ФЗ, однако и она не отвечает сложившейся на фармацевтическом рынке криминальной ситуации. Отсутствие единого понимания признаков преступлений в сфере оборота ФЛС, уголовно–правового определения понятия «фальсифицированное лекарственное средство» порождает не только трудности учета и контроля за распространенностью данных преступлений, но и проблему эффективного уголовно–правового противодействия рассматриваемому виду преступлений.

Большинство подделок приходится на лекарственные средства средней ценовой группы, часто используемые для лечения. Считается, что фальсифицировать дорогостоящие лекарства невыгодно, поскольку их медленнее раскупают. Основной недостаток приведенных оценок – их весьма условный характер. В действительности на основе приведенных цифр невозможно составить ясного представления о том, каковы абсолютные объёмы фальсифицированной лекарственной продукции, а самое главное, какова доля ФЛС в общем объеме оборота лекарств в России.

Оборот ФЛС приобретает повышенную степень общественной опасности в условиях рыночной экономики, подрывая устойчивость отечественных производителей и затрудняя регулирование системы лекарственного обеспечения населения. Оборот ФЛС является транснациональным деянием с присущими ему особенностями, среди которых отмечается: высокая прибыльность данного вида деятельности; постоянное увеличение доходов, в результате осуществления данного вида незаконной деятельности; широкомасштабная незаконная деятельность, контроль над огромными финансовыми средствами; применение коррупции для достижения основных целей.

Причины, обуславливающие оборот ФЛС, разнообразны. Сюда входят как общие причины: сложившаяся нестабильная социально-экономическая обстановка в стране, рост цен, невыплаты заработной платы, безработица, так и специфические причины, конкретно связанные с оборотом ФЛС: несовершенство нормы, предусматривающей уголовную ответственность за совершение оборота ФЛС. Для Российского фармацевтического рынка характерны основные черты развития мирового фармацевтического рынка, однако он имеет свои особенности развития, обусловленные новыми организационными, правовыми и экономическими, нравственно-психологическими причинами:

- бурного развития частного предпринимательства;
- низким уровнем отечественных фармацевтических производственных предприятий;
- юридической и экономической самостоятельностью аптечных организаций;
- большого числа иностранных производителей и фирм, занимающихся реализацией лекарственных средств;
- бесконтрольного увеличения оптовых и розничных организаций товаропроводящей сети;

- отсутствия льготных финансово–кредитных систем фармацевтических организаций;
- отнесения фармацевтического рынка к числу рынков с развитой конкуренцией, что обусловлено большим объемом и ассортиментом лекарственных средств.

1.3 Государственное регулирование фармацевтической отрасли

Современная экономика представляет собой синтез рыночного механизма и элементов государственного регулирования. В настоящее время нормальное функционирование любого рынка невозможно без вмешательства государства. Фармацевтический рынок Российской Федерации обеспечивает население страны социально значимой продукцией.

Увеличению емкости фармацевтического рынка способствуют несколько факторов. Во-первых, это общий рост заболеваемости из-за усиливающегося влияния техногенных факторов и ухудшения экологической обстановки. Во-вторых, это тенденция «старения населения» в развитых странах. В-третьих, в Китае, Индии, России, странах Восточной Европы, рост уровня доходов населения приводит к использованию более дорогостоящих и качественных препаратов.

Растет фармацевтический рынок также благодаря быстрому развитию таких сравнительно новых направлений как производство биологически активных добавок и дженериков — дешевых аналогов известных лекарств. С точки зрения системного подхода фармацевтический рынок Российской Федерации может быть представлен как совокупность взаимодействующих субъектов и объектов, а также воздействующих на них факторов. При этом субъектами будут являться участники рынка активные по отношению к объектам, на которые они воздействуют. Необходимо отметить, что под объектами фармацевтического рынка подразумеваются фармацевтические

товары, фармацевтическая продукция, вкусы и предпочтения потребителей, платежеспособная потребность, качество продукции, технологии и др.

Сфера производства и обращения фармацевтической продукции, также, требует особого отношения и контроля со стороны государства. Спецификой фармацевтического производства является выпуск особого продукта, направленного на улучшение или сохранение здоровья человека. Это обуславливает стратегическую роль фармацевтической промышленности для страны и важность государственного регулирования процесса производства лекарственных средств.

С учетом особенностей и специфики лекарственных средств как особого товара, непосредственно влияющего на состояние здоровья нации, на обращение лекарственных средств налагаются определенные ограничения, выражающиеся в необходимости обязательного лицензирования субъектов института обращения лекарственных средств, особом порядке допуска к фармацевтической деятельности и рядом других ограничений. Указанные ограничения, в целом, представляются оправданными спецификой рассматриваемой сферы деятельности и задачами государства по обеспечению должного контроля за качеством, эффективностью и безопасностью лекарственных средств.

Эффективная система безопасности служит фильтром от некачественных, контрафактных и фальсифицированных препаратов. Государство, регулирует деятельность субъектов производства и распределения, применяя такие механизмы, как лицензирование фармацевтической деятельности и производства лекарственных средств, регистрация лекарственных средств и регулирование цен на фармацевтическую продукцию. Кроме того, государственный контроль при обращении лекарственных средств осуществляется посредством проверки соблюдения правил доклинических и клинических исследований, лабораторной практики, организации производства и контроля качества лекарственных средств, оптовой торговли лекарственными средствами,

отпуска лекарственных препаратов, хранения и уничтожения лекарственных средств. В настоящее время одним из актуальных вопросов, требующих решения, является переход отечественных фармацевтических предприятий на стандарты GMP, которые представляют собой систему норм, правил и указаний в отношении производства лекарственных средств, медицинских устройств, изделий диагностического назначения, продуктов питания, пищевых добавок и активных ингредиентов.

Стандарт GMP отражает целостный подход к решению проблемы качества выпускаемой фармацевтической продукции, поскольку регулирует и оценивает собственно параметры производства и лабораторной проверки. Целью GMP является обеспечение наибольшей безопасности, идентичности и эффективности действия производимых лекарств.

Данный стандарт кардинально отличается от процедуры контроля качества путем исследования выборочных образцов продуктов, которая обеспечивает пригодность к использованию лишь самих этих образцов (и, возможно, партий, изготовленных в ближайшее к данной партии время).

Правила GMP устанавливают требования к системе управления качеством, контролю качества, персоналу, помещениям и оборудованию, документации, производству продукции, порядку отзыва продукции и организации самоинспекций. Стандарт распространяется на все виды ЛС и устанавливает общие требования к их производству и контролю качества, а также специальные требования к производству отдельных видов ЛС.

Роль государства в сфере обращения лекарственных средств очень велика, и государственное регулирование отношений осуществляется по нескольким направлениям: государственная регистрация лекарственных средств; лицензирование отдельных видов деятельности в сфере обращения лекарственных средств; аттестация и сертификация специалистов, работающих в этой сфере; осуществление государственного контроля производства, изготовления, качества, эффективности и безопасности

лекарственных средств; государственное регулирование цен на лекарственные средства.

Изменения, коснувшиеся системы государственной регистрации, были связаны с необходимостью исключения прямого контакта между заявителем (фармацевтической компанией) и экспертным учреждением, выдающим свое заключение при подаче документов на регистрацию.

Сейчас все производители фармацевтической продукции предоставляют заявление о государственной регистрации своих лекарств непосредственно в Министерство здравоохранения и социального развития. Государственная регистрация лекарственных средств обязательна в отношении:

- Оригинальных лекарственных препаратов и дженериков;
- Новых комбинаций зарегистрированных ранее лекарственных препаратов;
- Лекарственных препаратов, зарегистрированных ранее, но произведенных в других лекарственных формах, в новой дозировке;
- Производимых в России и импортируемых лекарственных средств.

Государственная регистрация лекарственных препаратов осуществляется по результатам экспертизы ЛС и этической экспертизы возможности проведения клинического исследования лекарственного препарата для медицинского применения.

Задание на проведение данных экспертиз выдается Министерством здравоохранения и социального развития. Учреждения, занимающиеся экспертизой лекарств, перешли от Росздравнадзора и Роспотребнадзора непосредственно к Минздравсоцразвития и были объединены в общую структуру.

Экспертиза лекарственных средств проводится комиссией экспертов экспертного учреждения, каждый из членов которой независимо и самостоятельно проводит исследования, оценивает результаты, полученные

им лично и другими экспертами, и формулирует выводы относительно поставленных вопросов в пределах своих специальных знаний.

Этапы государственной регистрации лекарственных препаратов можно представить следующим образом:

- Подача заявления о государственной регистрации лекарственного средства;
- Задание на проведение экспертизы;
- Экспертиза документов, выдача разрешения на клиническое исследование;
- Клиническое исследование и этическая экспертиза;
- Государственная регистрация, включение в реестр, выдача регистрационного удостоверения;

Таким образом, в результате анализа системы государственного регулирования деятельности субъектов фармацевтического рынка Российской Федерации, было выявлено, что государственное регулирование фармацевтического рынка базируется на соблюдении иерархии норм при использовании правовых, административных и экономических методов.

Регистрация ЛС является одним из компонентов системы защиты здоровья и жизни населения от небезопасных, неэффективных и некачественных фармацевтических препаратов. Регистрационная система фармацевтического рынка в последние годы подверглась значительным реформам.

В настоящее время государственная регистрация лекарственных препаратов осуществляется в соответствии с принятым в 2010 году федеральным законом РФ № 61-ФЗ «Об обращении лекарственных средств». Данный закон кардинально изменил порядок государственной регистрации.

1.4 Постановка задачи

В Российской медицине, главной проблемой является наличие некачественных или просроченных товаров. По статистике министерства здравоохранения попадание на прилавок аптек фальсификатов составляет немалый процент, что приводит к ухудшению здоровья, вплоть до летального исхода, а также сама аптека несет финансовые убытки из-за такой продукции, несмотря на то, что государство ужесточило контроль за производством и сертификацией лекарственных товаров. К сожалению, по штрих – коду, который уже имеется на упаковке лекарственных средств можно узнать небольшое количество информации о производителе.

В связи с выявленной проблемой было принято решение о создании интеллектуального метода распознавания текстовой информации на упаковках товаров, основываясь на модели искусственной нейронной сети. С помощью которой предоставляется возможность распознавать серию и дату на упаковке, а также возможность сравнивать полученные результаты со значениями, хранящимися в базе данных аптечной компании.

Целью работы является – «Разработка интеллектуального метода, основанного на аппарате искусственной нейронной сети для распознавания текстовой информации с упаковок товаров, для выявления, просроченного и фальсифицированного лекарственного средства».

Для решения данной цели сформулируем следующие задачи:

- Анализ методов распознавания образов;
- Исследование архитектуры нейросети;
- Проектирование и разработка нейросети.
- Обучение выбранной искусственной нейросети;
- Тестирование нейросети.

Выводы по разделу

В данном разделе был проведен анализ рынка лекарственных препаратов. При рассмотрении оборота фальсифицированных лекарств, стало очевидным, что главной проблемой рынка является некачественные препараты. Со стороны государства, регулировка ФЛС происходит в соответствии с законами РФ. Но проверки происходят не часто, что позволяет недобросовестным компаниям распространять нелегальный товар. На основании вышеизложенного, можно сделать выводы о том, что основной интенцией данной работы является уменьшение длительности обработки упаковок в аптечных пунктах, убрать ФЛС с прилавков аптек, а также выявить точные нормативные инструменты для оперативной корректировки и переформирования товарооборота в связи с изменением производственных ситуациях

2 Методы и модели распознавания текстовой информации

2.1 Классические методы распознавания образов

В настоящее время существует немало задач, в которых необходимо принять некоторое решение в зависимости от присутствия на изображении объекта или классифицировать его. Способность «распознавать» считается основным свойством биологических существ, в то время как компьютерные системы только начали развиваться, поэтому на сегодняшний день все же уступают в этом свойстве живому организму.

Рассмотрим общие элементы модели классификации.

Класс – множество объектов имеющие общие свойства. Для объектов одного класса предполагается наличие «схожести».

Классификация – процесс назначения меток класса объектам, согласно некоторому описанию свойств этих объектов.

Классификатор – устройство, которое в качестве входных данных получает набор признаков объекта, а в качестве результата выдающий метку класса.

Под образом будем понимать наименование области в пространстве признаков, в которой отображается множество объектов или явлений материального мира.

Признак – количественное описание того или иного свойства исследуемого предмета или явления.

Пространство признаков – это N-мерное пространство, определенное для данной задачи распознавания, где N – фиксированное число измеряемых признаков для любых объектов. Вектор из пространства признаков x , соответствующий объекту задачи распознавания это N-мерный вектор с компонентами (x_1, x_2, \dots, x_N) , которые являются значениями признаков для данного объекта.

Примерами задач классификации являются:

- распознавание символов;
- распознавание речи;
- установление медицинского диагноза;
- прогноз погоды;
- распознавание лиц и др.

Выделение характерных признаков или свойств из исходных изображений. Признак должен представлять из себя характерное свойство конкретного класса, при этом общие для этого класса. Признаки, характеризующие отличия между – межклассовые признаки. Выбор признаков является одной из важных задач, связанных с построением системы распознавания.

После того, как определены признаки необходимо определить оптимальную решающую процедуру для классификации. Рассмотрим систему распознавания образов, предназначенную для распознавания различных M классов, обозначенных как (x_1, x_2, \dots, x_m) . Тогда можно считать, что пространство образов состоит из M областей, каждая содержит точки, соответствующие образом из одного класса. Тогда задача распознавания может рассматриваться как построение границ, разделяющих M классов, исходя из принятых векторов измерений.

Решение задачи предварительной обработки изображения, выделение признаков и задачи получения оптимального решения и классификации обычно связано с необходимостью произвести оценку ряда параметров. Это приводит к задаче оценки параметров. Сравнение объектов можно производить на основе их представления в виде векторов измерений. Данные измерений удобно представлять в виде вещественных чисел. Тогда сходство векторов признаков двух объектов может быть описано с помощью евклидова расстояния.

$$\|x_1 - x_2\| = \sqrt{\sum_{t=1,d} (x_1 [i] - x_2 [i])^2} \quad (2.1)$$

где d – размерность вектора признака.

Разделяют 3 группы методов распознавания образов:

- Сравнение с образцом. В эту группу входит классификация по ближайшему среднему, классификация по расстоянию до ближайшего соседа. Также в группу сравнения с образцом можно отнести структурные методы распознавания. [32]

- Статистические методы. Как видно из названия, статистические методы используют некоторую статистическую информацию при решении задачи распознавания. Метод определяет принадлежность объекта к конкретному классу на основе вероятности. Примером служит метод на основе байесовского решающего правила.

- Нейронные сети.

Отдельный класс методов распознавания. Отличительной особенностью от других является способность обучаться.

В классическом подходе распознавания образов, в котором неизвестный объект для классификации представляется в виде вектора элементарных признаков. Система распознавания на основе признаков может быть разработана различными способами. Эти векторы могут быть известны системе заранее в результате обучения или предсказаны в режиме реального времени на основе каких-либо моделей.

Простой алгоритм классификация по ближайшему среднему значению заключается в группировке эталонных данных класса с использованием вектора математического ожидания класса (среднего значения).

$$\bar{x}_t = \frac{1}{n_i} \sum_{j=1, n_i} x_{i,j} \quad (2.2)$$

где $x_{i,j}$ – j -й эталонный признак класса i , n_j – количество эталонных векторов класса i .

Тогда неизвестный объект будет относиться к классу i , если он существенно ближе к вектору математического ожидания класса i , чем к векторам математических ожиданий других классов. Этот метод подходит для задач, в которых точки каждого класса располагаются компактно и далеко от точек других классов. Пример данного метода представлен на рисунке 2.1



Рисунок 2.1 – График группировки эталонных данных класса, с использованием вектора среднего значения

Проблема данного подхода заключается в том, что современные решаемые задачи основываются на классах, которые имеют несколько более сложную структуру как на рисунке 2.2. В данном случае класс 2 разделен на два непересекающихся участка, которые плохо описываются одним средним значением. Также класс 3 слишком вытянут, образцы 3-го класса с большими значениями координат x_2 ближе к среднему значению 1-го класса, нежели 3-го.

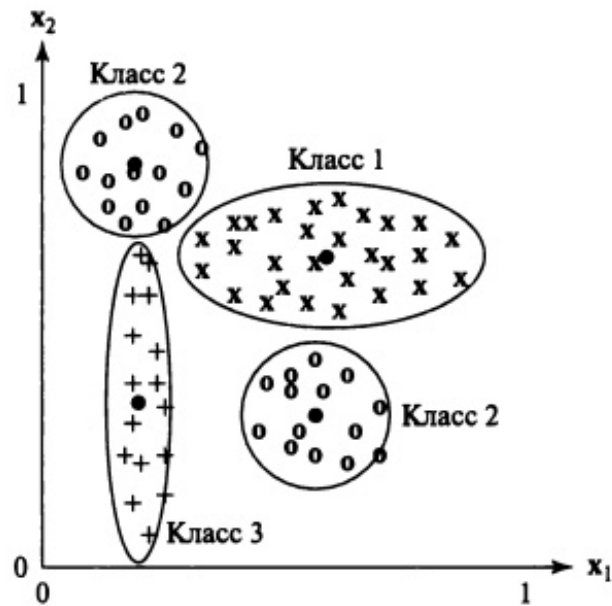


Рисунок 2.2 – Пример сложной группировки эталонных данных класса с использованием вектора среднего значения.

Другой подход классификации заключается в отнесении неизвестного вектора признаков x к тому классу, к отдельному образцу которого этот вектор наиболее близок. Это правило называется правилом ближайшего соседа. Классификация по ближайшему соседу может быть более эффективна, даже если классы имеют сложную структуру или, когда классы пересекаются.

При таком подходе не требуется предположений о моделях распределения векторов признаков в пространстве. Алгоритм использует только информацию об известных эталонных образцах. Метод решения основан на вычислении расстояния x до каждого образца в базе данных и нахождения минимального расстояния. Преимущества такого подхода очевидны:

- в любой момент можно добавить новые образцы в базу данных;
- древовидные и сеточные структуры данных позволяют сократить количество вычисляемых расстояний.

Кроме того, решение будет лучше, если искать в базе не одного ближайшего соседа, а k . Тогда при $k > 1$ обеспечивает наилучшую выборку

распределения векторов в d -мерном пространстве. Однако эффективное использование значений k зависит от того, имеется ли достаточное количество в каждой области пространства. Путем исследований стало очевидно, что в решаемой задаче больше двух классов, и принять верное решение оказывается невозможным.

Структурный (синтаксический) подход применяется к задачам распознавания образов, в которых важна информация о структуре конкретного объекта. В частности, такой метод распознавания, можно использовать для классификации символов алфавита, когда каждая буква может иметь различные варианты начертания, но в общем виде она будет выглядеть одинаково.

В рамках этого подхода считается, что образы состоят из различных подобразов, также как фразы и предложения строятся путем соединения слов.

Язык описания образов – язык, который обеспечивает структурное описание образов в терминах множества неприводимых элементов и операций композиции этих элементов.

На рисунке 2.3 показан пример структурного распознавания двух различных геометрических фигур. В рамках такого подхода распознавания образов считается, что они состоят из отдельных подобразов, соединенных различными способами.

Во многих задачах принятия решений априорная вероятностная информация о состояниях природы может быть изменена после получения новых экспертных оценок. Если рассматривать векторное представление образов, то в общем случае, вектор признаков состоит из компонент признаков, каждый из которых и их совокупность в целом характеризуют образ с той или иной степенью определенности. Эта неопределенность может, в частности, носить вероятностный характер.

Появление того или иного образа является случайным событием и вероятность этого события можно описать с помощью закона распределения

вероятностей многомерной случайной величины ξ в той или иной форме. Вид и параметры функции плотности определяются конкретной средой, в которой

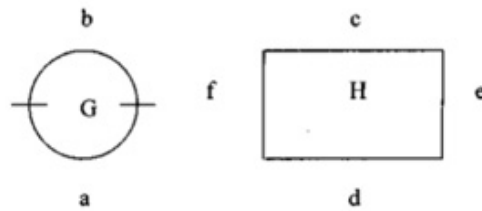


Рисунок 2.3 – Пример структурного распознавания двух различных геометрических фигур.

работает система распознавания. Зная элементы обучающей выборки можно восстановить вероятностные характеристики этой среды

Байесовский классификатор на основе наблюдаемых признаков относит объект к классу, к которому этот объект принадлежит с наибольшей вероятностью. Предположим, что мы наблюдаем некоторую случайную величину X , которая имеет плотность вероятности $p(x, w)$ с параметром w . А хотим сделать вывод о другой случайной величине w , имеющей некоторое распределение вероятности $\tau(w)$. Пусть в результате наблюдений получены статистические данные x . Из определения условной вероятности следует:

$$\Pr(w|x) = \frac{\Pr(x, w)}{\Pr(x)} \quad (2.3)$$

Также следует

$$\Pr(x, w) = \Pr(x|w) \Pr(w) \quad (2.4)$$

Подставив (2) в (1) получим формулу Байеса \Pr

$$\Pr(w|x) = \frac{\Pr(x|w) \Pr(w)}{\Pr(x)} \quad (2.5)$$

Для нашей задачи имеется m классов, т.е. m возможных переменных (w_1, \dots, w_m) , то

$$\Pr(w_i|x) = \frac{\Pr(x|w_i) \Pr(w_i)}{\Pr(x)} = \frac{\Pr(x|w_i) \Pr(w_i)}{\sum_{k=1..m} \Pr(w_k) \Pr(x|w_k)} \quad (2.6)$$

Распределение $\Pr(w)$ называют априорным распределением вероятностей возможных значений w . Это распределение принимается прежде, чем получены статистические данные.[28][29][31]

Распределение $\Pr(w|x)$ – апостериорное распределение значений w , при условии получения статистических данных. При этом w_i – гипотеза, а x – свидетельство, поддерживающее гипотезу. Если все классы w_i характеризуются собственными несовместными вероятностями, охватывающими все возможные случаи, то можно применить правило Байеса для вычисления апостериорных вероятностей каждого класса по априорным вероятностям этих классов и распределениям условной вероятности для x .

На практике необходимо каким-то образом обеспечить вычисление $p(x|w_i)$. Эмпирический подход предполагает разбиение диапазона x , на интервалы, затем подсчет частот появления x среди значений каждого интервала и сохранение в гистограмму. Необходимо нормировать функцию вероятности так, чтобы сумма всех значений x равнялась 1. Если x похоже на

одну из известных параметрических моделей, то можно представить распределение с помощью небольшого количества характеризующих его параметров.

Обычно используется распределения Пуассона, экспоненциальное и нормальное.

Непрерывная случайная величина X имеет нормальный закон распределения (закон Гаусса) с параметрами μ и σ^2 , если ее плотность вероятности имеет вид:

$$p(x) = N(\mu, \sigma)(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.7)$$

Пример распределения Гаусса для различных параметров μ и σ показан на рисунке 2.4.

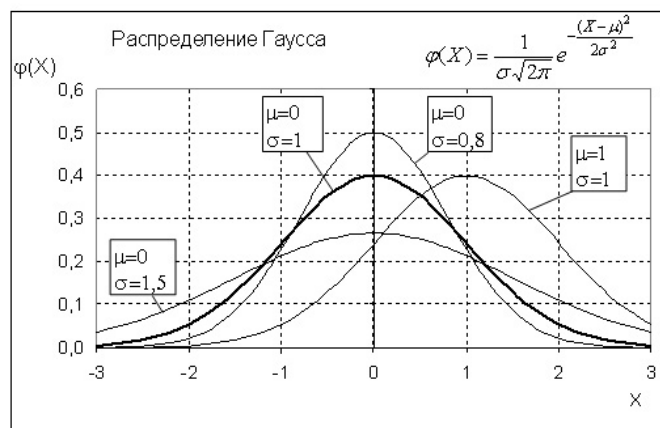


Рисунок 2.4 – Пример распределения Гаусса для различных параметров μ и σ

Применительно к классификатору: вычисляются расстояния между векторами или вероятности $P(w_i|x)$, что по правилу Байеса можно вычислить как $p(x|w_i)P(w_i)/p(x)$. $p(x)$ – одинакова для всех функциональных блоков, следовательно, ее можно игнорировать и принимать классификационное решение w_i по максимальному значению $p(x|w_i)P(w_i)$. Для проектирования

байесовского классификатора необходимы сведения Y об априорной вероятности каждого класса $p(x|w_i)$. Знание этих характеристик позволяет принимать оптимальные решения в процессе распознавания. Из вышесказанного следует, что такой подход для вычисления априорной вероятности в нашем конкретном случае трудно оценить, в связи с этим будет трудно и составить математическую модель основываясь на вероятности.

Нейронная сеть имитирует структуру и свойства организации нервной системы живых организмов: она состоит из большого числа соединённых между собой простых однотипных вычислительных элементов (нейронов) и, как система, обладает более гибким и сложным поведением по сравнению с возможностями каждого отдельного нейрона. Нейросеть получает на входе набор входных сигналов и выдает соответствующий им ответ (выходные сигналы), являющийся решением некоторой задачи.

Каждый нейрон выполняет небольшой объем работ – например, суммирует пришедшие на него сигналы с некоторыми весовыми коэффициентами и затем дополнительно нелинейно преобразует эту взвешенную сумму входных данных.

Нейроны группируются в последовательность слоёв; входные сигналы поступают на первый слой и последовательно проходят через все слои до последнего, выходного слоя сети. Но бывают и рекуррентные структуры, обеспечивающие циркуляцию некоторого набора внутренних сигналов. Пример нейронной сети можно увидеть на рисунке 2.5.

В общем случае поток данных (сигналов) может формировать и перекрёстные, и обратные связи. Структура нейросети может быть адаптирована к задаче. В ИНС могут быть включены дополнительные нейроны и даже слои нейронов, если исходно она была не способна обеспечить нужную точность решения. Из нейросети могут быть исключены лишние нейроны и связи между ними, если исходная сеть была избыточна. Нейросеть может сама выделить наиболее информативные для задачи входные сигналы, отбросить неинформативные, шумовые сигналы и в итоге повысить надежность решения. При этом

коррекция размеров нейронной сети не приводит к полному забыванию ранее сформированных при обучении навыков, что ускоряет последующий процесс дообучения нейросети.

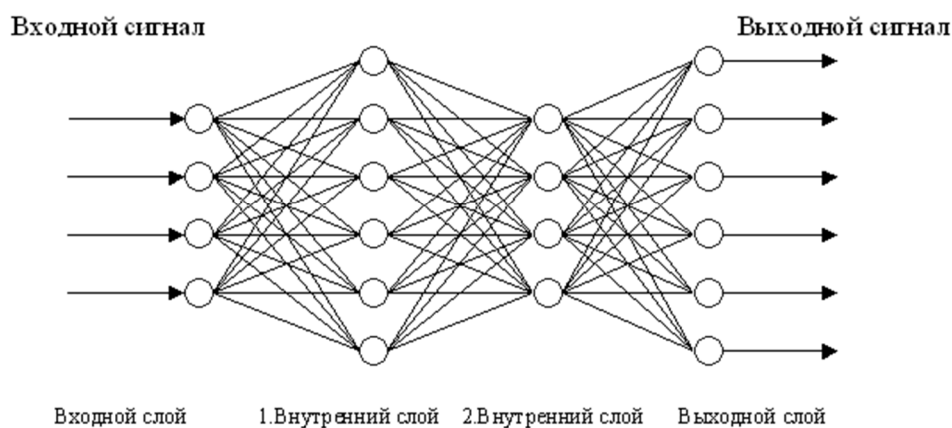


Рисунок 2.5 – Пример нейронной сети с 2-мя скрытыми слоями

Искусственная нейросеть, как и её биологический прототип, может обучаться: она содержит внутренние адаптивные параметры нейронов и своей структуры, и, меняя их, может менять свое поведение, добиваясь улучшения точности решения некоторой задачи.[20][32][41]

По сравнению с линейными методами статистики (линейная регрессия, авторегрессия, линейный дискриминант) нейросети позволяют эффективно строить нелинейные зависимости, более точно описывающие наборы данных. Из нелинейных методов классической статистики известен байесовский классификатор, строящий квадратичную разделяющую поверхность – а искусственная нейронная сеть может построить поверхность более высокого порядка. Высокая нелинейность разделяющей поверхности наивного байесовского классификатора (он не использует ковариационные матрицы классов, как классический Байес, а анализирует локальные плотности вероятности) требует значительного суммарного числа примеров для возможности оценивания вероятностей при каждом сочетании интервалов

значений переменных – тогда как нейросеть обучается на всей выборке данных, не фрагментируя её, что повышает адекватность настройки сети.

Необходимость решения прямой и обратной задач обычно требует построения двух моделей. При использовании же нейронных сетей можно обойтись одной сетью, обученной решать прямую задачу. Так же можно отметить что нейросеть одновременно может решать несколько задач при наличии у нее нескольких выходов. Это дает преимущество перед структурными и классифицированными методами.

2.2 Нейросетевые методы распознавания информации

Как уже было описано ранее ИНС представляет собой совокупность нейронов, которые составляют слои. В каждом слое нейроны между собой никак не связаны, но связаны с нейронами предыдущего и следующего слоев.

Количество слоев и нейронов в них определяют точность и достоверность получаемых результатов при решении задач, т. е. чем больше слоев и нейронов на каждом слое – тем меньше ошибок и выше надежность работы сети. Однако, если построить слишком большую сеть, то можно столкнуться с уменьшением производительности и увеличением сложности модели. Поэтому для решения неоднотипных задач используется разная архитектура искусственной нейронной сети. [14][30][37]

На данный момент времени существуют три основных архитектуры ИНС для распознавания информации: многослойный персептрон, сеть радиально-базисных функций (РБС) и свёрточная нейронная. Их так же используют для классификации образов, прогнозирования, управление и т.д.

В середине 20-х годов интерес к нейронным сетям возрос. Многие ученые пытались улучшить основные свойства сети и нейрона в частности, ввиду того, что имеющиеся архитектуры отставали от желаемых возможностей. Первым достичь успехов в этой области удалось сделать Фрэнку Розенблатту в 1958 году. Он предложил математическую, а затем и

электронную модель многослойного персептрона. Многослойными персептронами называют нейронные сети прямого распространения. Входной сигнал в таких сетях распространяется в прямом направлении, от слоя к слою. Многослойный персептрон в общем представлении состоит из следующих элементов:

- множества входных узлов (вектор x), которые образуют входной слой. Входной слой не выполняет никаких вычислений – он просто распределяет вектор x на следующий слой и служит приемником информации;
- одного или нескольких скрытых слоев вычислительных нейронов, в котором и происходят расчеты;
- одного выходного слоя нейронов, который выдает результат расчетов сети (OUT).

Количество входных и выходных элементов в многослойном персептроне определяется условиями задачи. Основной проблемой для построения персептрона является неизвестность использования промежуточных слоев и элементов в них. В качестве начального приближения можно взять один промежуточный слой, а число элементов в нем положить равным полусумме числа входных и выходных элементов.

Пример многослойного персептрона можно наблюдать на рисунке 2.6

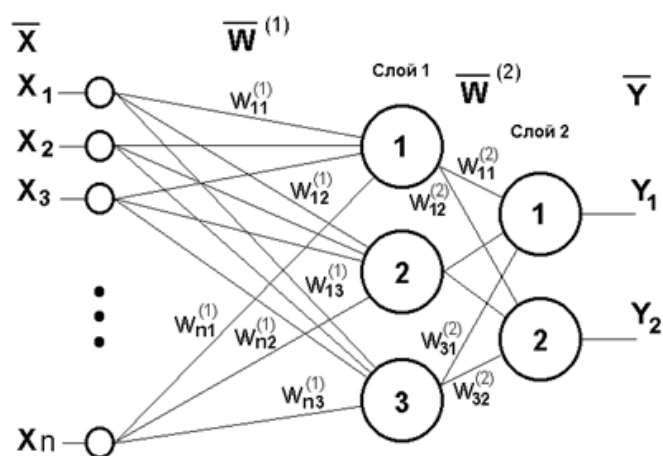


Рисунок 2.6 – Пример модели многослойного персептрона

Работа многослойного персептрона описывается следующими формулами:

$$NET_{ji} = \sum_i w_{ijl} x_{ijl} \quad (2.8)$$

$$OUT_{ji} = F(NET_{ji} - \theta_{ji}) \quad (2.9)$$

$$x_{ij} = OUT_{ij} \quad (2.10)$$

где индексом i обозначим номер входа, j – номер нейрона в слое, l – номер слоя.

X_{ijl} – j -й входной сигнал j -го нейрона в слое l ;

W_{ijl} – весовой коэффициент i -го входа нейрона номер j слоя l ;

NET_{ij} – сигнал нейрона номер j слоя l ;

F – нелинейная функция, называемая функцией активации.

θ – пороговый уровень данного нейрона;

Введем обозначения: W_{jl} – вектор-столбец весов для всех входов нейрона j в слое l . W_j – матрица весов всех нейронов в слое j . В столбцах матрицы расположены вектора W_{jl} . Аналогично X_{jl} – входной вектор-столбец слоя l .

Многослойная нейронная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов. Многослойные нейронные сети являются универсальным аппроксиматором функций. Общая формула, описывающая многослойный персептрон:

$$f(x) = F \left(\underbrace{\sum_{i_N} w_{i_N j_N N} \dots \sum_{i_2} w_{i_2 j_2 2} F \left(\underbrace{\sum_{i_1} w_{i_1 j_1 1} - \theta_{i_1 1}}_{\text{слой 1}} \right) - \theta_{j_2 2} \dots - \theta_{N_2 N}}_{\text{слой 2}} \right) \quad (2.11)$$

$\text{слой } N$

Для распознавания информации на упаковках товаров основной причиной не эффективного построения данной архитектуры нейронной сети является распределенная форма нелинейности и высокая связность сети, которые существенно усложняют теоретический анализ многослойного персептрона.

Рассмотрим другой метод распознавания информации нейронных сетей, называемый нейронными сетями с радиально-базисными функциями. Нейронная сеть радиальных базисных функций содержит в себе наиболее простой форме три слоя: обычный входной слой, выполняющий распределение данных образца для первого слоя весов; слой скрытых нейронов с радиально симметричной активационной функцией, каждый j -й из которых предназначен для хранения отдельного эталонного вектора в виде вектора весов $W_j^{(h)}$; выходной слой.

Для построения сети РБС необходимо выполнение следующих условий:

- наличие эталонов, представленных в виде весовых векторов нейронов скрытого слоя;
- наличие способа измерения расстояния входного вектора от эталона. Обычно это стандартное евклидово расстояние;
- Специальная функция активации нейронов скрытого слоя, задающая выбранный способ измерения расстояния. Обычно используется функция Гаусса, существенно усиливающая малую разницу между входным и эталонным векторами.

Радиально-базисную модель изображена на рисунке 2.7

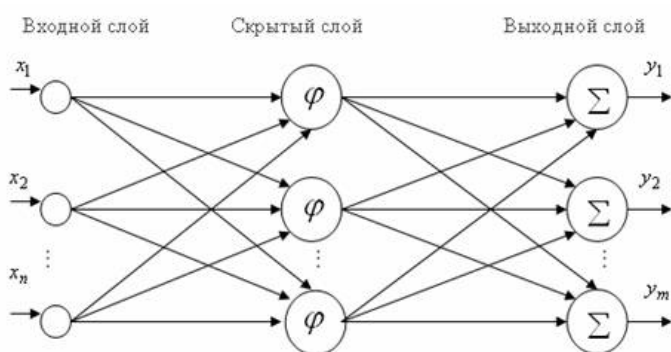


Рисунок 2.7 – Пример модели с радиальными базисными функциями

Выходной сигнал эталонного нейрона скрытого слоя a_j – это функция (гауссиан) только от расстояния p_j между входным и эталонным векторами:

$$p_j = \sqrt{\sum_{i=1}^n (x_i - w_{ij}^{(h)})^2} \quad (2.12)$$

$$a_j = e \left[- \left(\frac{p_j - R}{\sigma_j} \right)^2 \right]; j = 1, \dots, m_h \quad (2.13)$$

где $p_j = \eta_j$ – активационный уровень нейрона j скрытого слоя; $x = (x_1, \dots, x_n)^T$ – входной вектор; $W_j^{(h)} = (W_{1j}^{(h)}, \dots, W_{nj}^{(h)})^T$ – весовой вектор j -го эталонного нейрона скрытого слоя; R, σ_j – параметры активационной функции; m_h – число эталонов – нейронов скрытого слоя.

Таким образом, выходной сигнал шаблонного нейрона – это функция только от расстояния между входным вектором x и сохраненным центром $W_j^{(h)}$:

$$a_j = g_j(x) = \varphi \left(\frac{\|x - w_j^{(h)}\|}{\sigma_j} \right) \quad (2.14)$$

К недостаткам сетей сеть радиально-базисных функций можно отнести то, что заранее должно быть известно число эталонов, а также эвристики для построения активационных функций нейронов скрытого слоя, а также данные сети обладают плохими экстраполирующими свойствами и получаются весьма громоздкими при большой размерности вектора входов.[13][21][17]

Еще одним методом распознавания информации является свёрточная нейронная сеть(СНС), пример которой представлен на рисунке 2.8. СНС – специальная архитектура искусственных нейронных сетей, предложенная

Яном Лекуном и нацеленная на эффективное распознавание изображений. Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоев и субдискретизирующих, пример которых изображен на рисунке 2.8. Структура сети – однонаправленная (без обратных связей), принципиально многослойная. Функция активации нейронов – любая, по выбору исследователя.[27]

Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

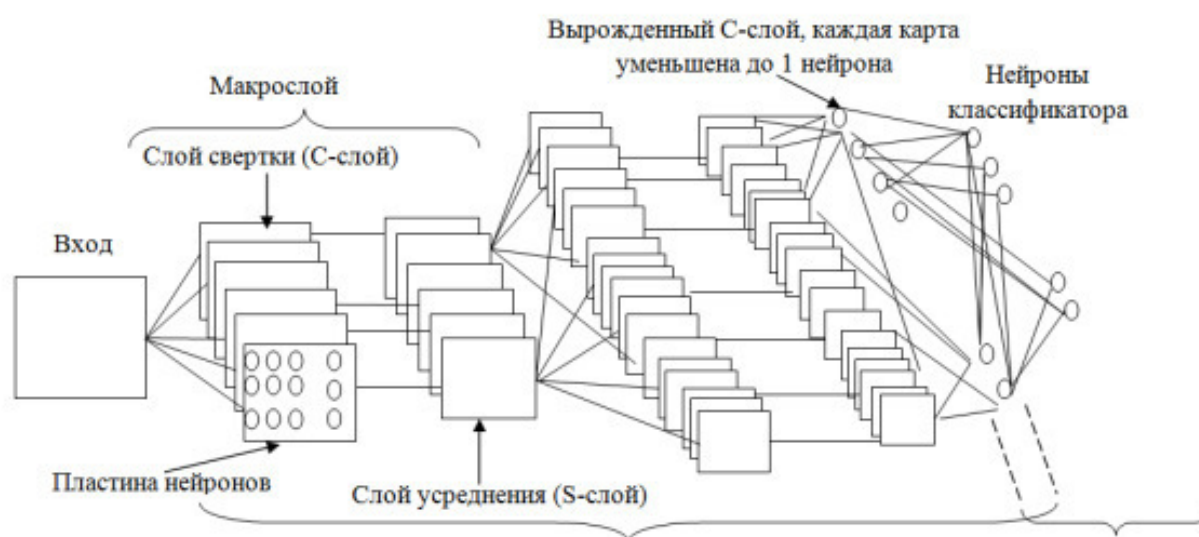


Рисунок 2.8 – Пример модели свёрточной нейронной сети

В обычном перцептроне, который представляет собой полносвязную нейронную сеть, каждый нейрон связан со всеми нейронами предыдущего слоя, причем каждая связь имеет свой персональный весовой коэффициент. В свёрточной нейронной сети в операции свёртки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою, формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются общие веса - матрица весов, которую также называют набором весов или ядром свёртки. Она построена таким образом, что графически кодирует какой-либо один признак.

Естественно, в свёрточной нейронной сети набор весов не один, а целая гамма, кодирующая всевозможные линии и дуги под разными углами. При этом такие ядра свертки не закладываются исследователем заранее, а формируются самостоятельно путём обучения. Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многомерной.

Операция субдискретизации выполняет уменьшение размерности сформированных карт признаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон карты признаков уменьшенной размерности.[5][9][33]

Таким образом, повторяя друг за другом несколько слоёв свёртки и субдискретизации строится свёрточная нейронная сеть.

Если на первом слое двумерное ядро свёртки проходит по одноканальному изображению, то на внутренних слоях, состоящих из множества карт признаков, берётся трёхмерное ядро размера $[K \times K \times N]$, где K - размер ядра, а N - количество карт признаков в текущем слое, формируя (после прохождения функции активации) одну карту признаков следующего слоя, соответствующую этому ядру свертки.

По сравнению с другими архитектурами нейросети является гораздо меньшее количество настраиваемых весов, так как одно ядро весов используется целиком для всего изображения, вместо того, чтобы делать для каждого пикселя входного изображения свои персональные весовые коэффициенты. Это подталкивает нейросеть при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждой показанной картинке в мириадах весовых коэффициентов, как это делает перцептрон. Так же у СНС удобное распараллеливание вычислений, а, следовательно, возможность реализации алгоритмов работы и обучения сети

на графических процессорах и относительная устойчивость к повороту и сдвигу распознаваемого изображения.

2.3 Выбор и обоснование архитектуры и алгоритма обучения искусственной нейронной сети

2.3.1 Анализ архитектур сверточных нейронных сетей

Как уже упоминалось выше, первой предложенной сверточной нейронной сетью была модель Яна Лекуна – LeNet, схема которой представлена на рисунке 2.9. Это была первая модель, которая содержала чередующиеся два раза сверточные слои и слои субдискретизации, а также три полносвязных слоя.

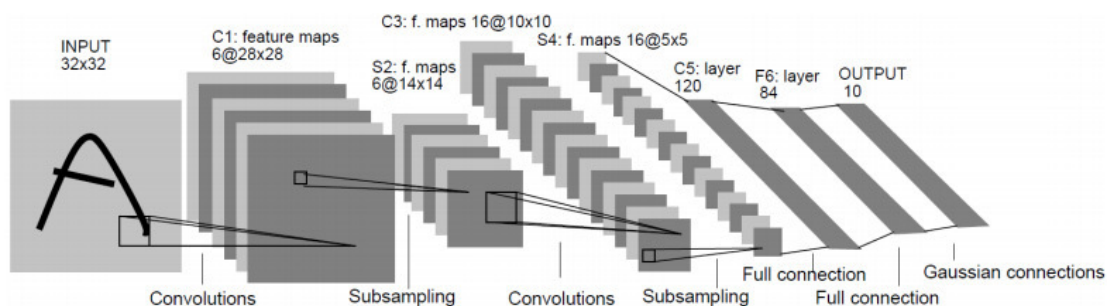


Рисунок 2.9 – Схема архитектуры сверточной нейронной сети LeNet

Данная архитектура без значительных изменений до сих пор в большинстве случаев используется при решении прикладных задач медицинской диагностики на изображениях и видеоданных. Следующей значительной сверточной нейронной сетью была сеть AlexNet, предложенная Алексом Крижевским (Alex Krizhevsky). Данная сеть была очень похожа на сеть LeNet, однако отличалась от неё более массивной и сложной архитектурой. Также она имела всего лишь один сверточный слой и несколько слоев субдискретизации по принципу выбора максимального значения. Схема архитектуры данной сети представлена на рисунке 2.10.

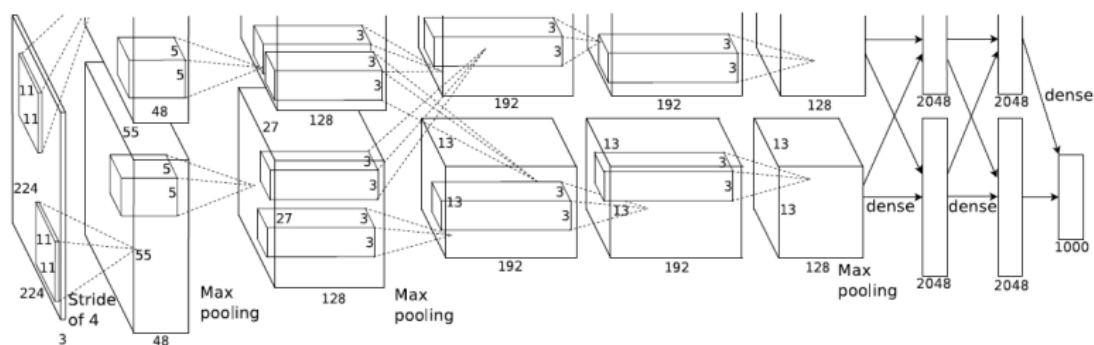


Рисунок 2.10 – Архитектура сети AlexNet

В работе, посвященной разработке данной сети также были предложены 96 фильтров для СНС, которые используются при проектировании сверточных сетей и в настоящее время – рисунок 2.11.

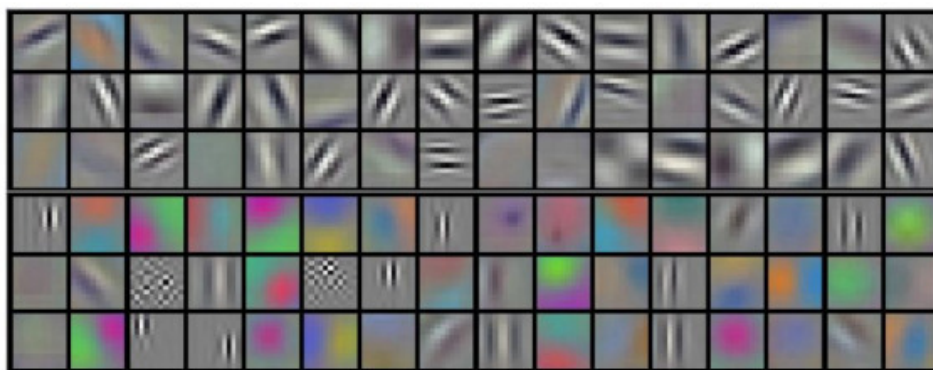


Рисунок 2.11 – Фильтры сети AlexNet Сверточная нейронная сеть

AlexNet предназначена для распознавания объектов любой сложности на больших изображениях размером 224 x 224. В 2012 году она стала победителем в конкурсе ImageNet, значительно обогнав своих конкурентов.

2.3.2 Выбор алгоритма сегментации изображения

Сегментацией изображения называют разбиение изображения на области или сектора, отличающиеся друг от друга по каким-либо признакам. При решении задач обработки изображений и компьютерного зрения,

сегментация играет важную роль. Область – это множество четырех или восьмисвязных пикселей при определении «соседства» в соответствии с выбранным признаковым пространством. [22][24][40]

Сегментация активно используется при обработке медицинских изображений, например, для обнаружения опухолей и других патологий, определение объёмов тканей, различных диагностик, планирование лечения; при выделении объектов на спутниковых снимках, распознавание лиц, распознавание отпечатков пальцев, в системах управления дорожным движением и т.д. А также для обработки объектов видеопоследовательности.

Это позволяет акцентировать внимание на существенных деталях и пренебрегать качеством несущественных. Выделение границ различными алгоритмами должны были бы приводить к одинаковому результату, но поскольку на сегментацию влияет большое количество различных факторов, ни один из алгоритмов не дает достаточно точной информации о границах и областях.

Сегментация обычно используется не самостоятельно, а как часть некоторой системы (например, системы машинного зрения), поэтому качество работы алгоритма оценивается исходя из работы системы в целом. Выделяют следующие алгоритмы сегментации:

- Пороговые алгоритмы;
- Алгоритм наращивания областей;
- Граничные алгоритмы;
- Алгоритмы сегментации на основе кластеризации.

Если необходимо распознавать изображение, области которого значительно различаются по яркости, то рекомендуется использовать пороговые алгоритмы, например, для сегментации алгоритмом порогового распределения необходимо получить бинарное изображение из полутонного. Для этого устанавливается некоторое пороговое ограничение. После

квантования функция изображения отображает элементы изображения с уровнем яркости больше порогового в значение 1, меньшее порогового 0.

В таких случаях оптимальное значение порога определяется достаточно легко на основе анализа гистограмм. Дальнейшее его естественное развитие – мультипороговая сегментация. При отсутствии шумов и других возмущений, колебание яркости происходит только на границах выделяемой области. Такая модель может быть упрощена до единой размерности, тогда распределение яркости однозначно совпадает с формой контура границы.

Данные алгоритмы позволяют сегментировать простые изображения, но, как правило, не дают хорошего результата на изображениях с наличием неровного освещения, вспышек, теней и различных помех, что, к сожалению, часто встречается в реальных ситуациях. Кроме того, при наличии значительных интервалов между экстремумами определение значимого минимума может быть затруднительным.

Для уменьшения влияния указанных недостатков разработаны алгоритмы, которые реализуют анализ взвешенных значений экстремумов, а точнее анализируют интенсивность и градиент. В общем случае, для корректного использования алгоритмов пороговых значений, нужно избегать «смещения» при выборе порогового значения путем жесткого контроля одинаковости распределения в темных и светлых областях гистограммы яркости; необходимо разбивать изображение на возможно малые элементы, чтобы гистограмма яркости имела ярко выраженные экстремумы; эти элементы должны быть достаточно большими, чтобы объем статистической выборки позволял удовлетворительно оценить местоположение экстремумов и описать окрестность.

В отличие от порогового алгоритма, у наращивания областей происходит группировка соседних элементов с одинаковыми или близкими уровнями яркости, которые после объединяются в однородные области. Наиболее распространенные алгоритмы нахождения областей включают в

себя такие алгоритмы, как слияние- расщепление областей, алгоритм водоразделов и алгоритм центроидного связывания.[7][8][10]

Более сложные алгоритмы построены на объединении примитивных областей. При центроидном связывании на основании информации об объекте выбираются стартовые точки, которым присваиваются различные метки. Точки с одинаковыми метками образуют отдельные множества. Такой алгоритм наиболее пригоден для сегментации простых изображений. Для более сложных изображений выбор точек проводится по итерациям, на каждой из которых рассматривается набор точек на предмет принадлежности их соседям данному множеству. Точки, включенные в множество на предыдущих итерациях, не рассматриваются. Так проводится анализ всех множеств по очереди. Точки, добавленные к множеству на данной итерации, называются фронтом, а объединение фронтов – волной, поэтому такие алгоритмы получили название – волновые. Алгоритм слияния-расщепления разделяет точки изображения путем разбиения некоторым образом изображения на квадраты, которые затем анализируются для их проверки на однородность (чаще всего это однородность по яркости). удовлетворяет условиям однородности, он заменяется четырьмя «подквadrатами», а подходящие по условию однородности 4 квадрата могут быть объединены в одну область.

К недостаткам алгоритмов нахождения областей в первую очередь относят то, что они выделяют общие фрагменты, зачастую не показывая информации об изменении яркости внутри области и возможных внутренних границах.

Другой способ сегментации заключается в нахождении точек, лежащих на границах областей. Характеристика яркости - основная для определения границ. Границей называют сильные локальные изменения яркости на изображении. Фактически они разделяют области. Часто граница определяется как скачок значения первой производной яркости изображения. Такие «разрывности» (неоднородности) могут быть пошаговыми, когда яркость на изображении резко меняет свое значение вплоть до

противоположного значения, и линейными, когда значение яркости также внезапно меняется, но через короткий промежуток возвращается к первоначальному состоянию.

Алгоритмы выделения границ состоят из фильтрации (улучшение выделения границ при наличии шумов), усиления (акцент на точках, где происходит перепад яркости), выделения (используя пороговое значение, решение, выделять ли точки в границу) и локализации (определение месторасположения и направления). Разработано много способов выделения границ, например, операторы Робертса, Собела, Превитта, Канни.

В целом, выделение границ дает весьма неплохие результаты для интерпретации изображения.

Однако, для решаемой проблемы наблюдается большая вычислительная сложность, что не позволяет использовать алгоритм. Необходимо учитывать также, что для точной оценки значения и направленности границ необходимо использовать разные маски; получение маски путем перестановки коэффициентов циклически по соседям «по квадрату» не будет гарантировать достоверный результат; оптимизация в целях устранения шумов накладывает на маски дополнительные условия.

Отдельной ветвью выделяются алгоритмы сегментации, основанные на кластеризации. Их преимущество - они автоматические и могут быть использованы для любого количества признаков и классов.

2.3.3 Выбор алгоритма обучения

Все алгоритмы локальной оптимизации можно разделить на два класса: алгоритмы локальной оптимизации с вычислением частных производных первого порядка и алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядков.

Алгоритм Левенберга-Марквардта предназначен для оптимизации параметров нелинейных регрессионных моделей. Предполагается, что в

качестве критерия оптимизации используется среднеквадратичная ошибка модели на обучающей выборке. Алгоритм заключается в последовательном приближении заданных начальных значений параметров к искомому локальному оптимуму.

Задана обучающая выборка – множество пар свободной переменной $x \in X^M$ (входы сети) и зависимой переменной $y \in Y^M$. Задана функциональная зависимость, представляющую собой регрессионную модель $y = f(w, x_n)$, непрерывно дифференцируемую в области $W \cdot X$. Параметр w является вектором весовых коэффициентов. Требуется найти такое значение вектора w , которое бы доставляло локальный минимум функции ошибки

$$E_D = \sum_{n=1}^N (y_n - f(w, x_n))^2 \quad (2.16)$$

Перед началом работы алгоритма задается начальный вектор весовых коэффициентов w .

1. На каждом шаге итерации этот вектор заменяется на вектор $w + \Delta w$.

$$J = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial w_1} & \dots & \frac{\partial f(w, x_1)}{\partial w_R} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(w, x_N)}{\partial w_1} & \dots & \frac{\partial f(w, x_N)}{\partial w_R} \end{bmatrix}$$

2. Для оценки приращения Δw используется линейное приближение функции $f(w + \Delta w, x) \approx f(w, x) + J \Delta w$ где J – якобиан функции $f(w, x_n)$ в точке w . Матрицу J наглядно можно представить в виде

Здесь вектор весовых коэффициентов $w = [w_1, \dots, w_R]^T$. Приращение в точке w , доставляющее минимум E_D равно нулю. Поэтому для нахождения последующего приращения Δw приравняем нулю вектор частных производных E_D по w .

$$E_D = |y - f(w + \Delta w)|^2 \text{ где } f(w + \Delta w) = [f(w + \Delta w, x_1), \dots, f(w + \Delta w, x_N)]^T$$

Преобразовывая и дифференцируя это выражение получим

$$|y - f(w + \Delta w)|^2 = (y - f(w + \Delta w))^T (y - f(w + \Delta w)) = f^T(w + \Delta w) f(w) - 2Y^T f(w + \Delta w) + Y^T y$$

Таким образом, чтобы найти значение Δw нужно решить систему линейных уравнений $\Delta w = (J^T J)^{-1} J^T (y - f(w))$. Так как число обусловленности матрицы $J^T J$ есть квадрат числа обусловленности матрицы J , то матрица $J^T J$

$$\frac{\partial E_D}{\partial w} = (J^T J) \Delta w - J^T (y - f(w)) = 0$$

может оказаться существенно вырожденной. Поэтому Марквардтом введен параметр регуляризации $\lambda \geq 0$.

3. $\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w))$ где I – единичная матрица. Этот параметр назначается на каждой итерации алгоритма. Если значение ошибки E_D убывает быстро, малое значение λ сводит этот алгоритм к алгоритму Гаусса-Ньютона.

4. Алгоритм останавливается в том случае, если приращение Δw в последующей итерации меньше заданного значения, либо если вектор весовых коэффициентов доставляет ошибку E_D , меньшую заданной величины, или если исчерпано число циклов обучения НС. Значение вектора w на последней итерации считается искомым. [23][19][16]

Метод обратного распространения ошибки, являющийся градиентным методом локальной оптимизации с вычислением производных первого порядка.

Алгоритм обратного применяется для обучения многослойных слоистых НС. Нейроны обычно имеют сигмоидальную функцию возбуждения. Обучение такой НС основано на вычислении отклонений выходных сигналов от эталонных и "прогоне" этих величин вплоть до входного слоя с целью коррекции ошибок. [25][34][35]

Алгоритм обратного распространения был изобретен в 1974 г. Полем Дж. Вербосом.

Согласно этому алгоритму вектор ошибки определяется как разность вектора выходов и эталонного вектора выходов. Подсчитывается квадрат модуля вектора ошибки, т.е. сумма квадратов разностей компонент указанных векторов. Затем меняются веса связей для минимизации ошибки. Каждый вес

меняется на величину, пропорциональную скорости, с которой изменяется ошибка по мере изменения веса.

Обозначим индексом j выходной слой, i – предыдущий слой. Суммарная ошибка вычисляется следующим образом:

$$E = \frac{1}{2} \sum_{k=1}^{N_j} (Y_{jk} - D_{jk})^2, \quad (2.17)$$

где Y_{jk} – выходная активность k -го нейрона j -го слоя;

D_{jk} – эталонный выход k -го нейрона j -го слоя.

Активность нейрона есть функция его входа, например сигма-функция:

$$Y_{jk} = \frac{1}{1 + e^{-X_{jk}}}, \quad (2.18)$$

где X_{jk} - суммарный взвешенный вход k -го нейрона j -го слоя, который вычисляется следующим образом:

$$X_{jk} = \sum_{l=1}^{N_i} (Y_{il} * W_{il}^{jk}), \quad (2.19)$$

где Y_{il} – выход l -го нейрона i -го слоя;

W_{il}^{jk} – вес связи;

Суммирование проводится по всем нейронам i -го слоя, связанным с данным нейроном j -го слоя.

Данная модель обучения подойдет для распознавания текста на упаковках товаров больше, чем алгоритм Левенберга-Марквардта. Главным его отличием стало опыт в положительном применении данного алгоритма к СНС, а также его простота организации и эффективной работы. Данный факт был выявлен путем сравнения двух алгоритмов обучения.[11][12][15]

Выводы по разделу

В данном разделе рассмотрены классические методы распознавания изображения. Наиболее подходящим способом для реализации поставленной задачи стала искусственная нейронная сеть, которая предполагает под собой 3 архитектуры: многослойный персептрон, радикально-базисная и сверточная сети. Сверточная сеть по результатам экспериментов показала себя эффективнее других, по реализации сети (основными критериями были наименьшее количество нейронов, не уменьшающих ее качество и скорость обучения). Соответствующим методом обучения был выбран метод обратного распространение ошибки.

3 Программная реализация нейросетевой архитектуры

3.1 Формирование сверточной архитектуры искусственной нейронной сети

В магистерской диссертации рассматривается сверточная нейронная сеть для распознавания изображений серии и даты на упаковке ЛС размерностью 12x18 пикселей.

Нейронная сеть состоит из следующих слоев:

- входной слой;
- свёрточный слой;
- субдискретизационный слой;
- сверточный слой;
- Нейросетевой слой с сигмоидальной функцией активации;
- Нейросетевой слой с гиперболическим тангенсом.

Структура нейронной сети выявлена путем проведения экспериментов. Увеличенное на одну строку и столбец начальное изображение, содержащееся в первом слое, подается на вход сверточному слою, состоящему из 30 карт признаков размером 6×8 каждая. Следовательно, сверточный слой насчитывает $6 \times 8 \times 30 = 1620$ нейронов. Далее обрабатывается только часть изображения с размером ядра 6×6 . Из чего можно заключить, что существующим картам признаков соответствует $6 \times 6 = 36$ весов и 1 смещение. В общей сложности, $(6 \times 6 + 1) \times 30 = 1110$ весов и $(6 \times 6 + 1) \times 1620 = 59940$ связей в сверточном слое. Поскольку нейрон из случайно выбранной карты признаков выполняет свертку с размером ядра 6×6 некой части изображения размером 6×6 пикселей, то изображение целиком разделяется на перекрывающиеся области 6×6 пикселей. Перекрытие равняется 4-м пикселям, из вышеперечисленного можно сделать вывод: по вертикали 8 областей 6×6 пикселей, по горизонтали 5 областей 6×6 пикселей, Итого,

размерность карты признаков – 5×8 нейронов. Далее полученные результаты подаются на полносвязный слой, который состоит из 110 нейронов, $110 \times (110 + 1) = 12210$ весов и связей (как уже было описано в разделе 2, в полносвязном слое каждый нейрон содержит в себе неповторимый весовой коэффициент). Следующий и последний слой, является полносвязным и включает в себя 14 нейронов ($110 \times 14 = 1540$ связей и весов). Далее получаются значения из 14 чисел вещественного значения в диапазоне $[-1, 1]$. В идеальном случае, все выходные значения должны быть равны -1, кроме значения, соответствующего классу входного изображения – оно должно быть равно 1.

3.2 Алгоритмы и технология решения задач

Программное средство будет выполнять следующие операции:

- Создать искусственную нейронную сеть;
- Обучить нейросеть;
- Тестировать нейросеть;
- Сравнить результаты обучения с имеющимися данными в БД.

Первым этапом данного алгоритма, является вход в систему, после которого, пользователю предоставляется возможность создать искусственную нейронную сеть. Если он выбирает создание нейросети, то у него есть возможность ввести необходимые данные и непосредственно создать ИНС. Но в случае, если у пользователя нет необходимости выполнять вышеизложенный этап, тогда ему предоставляется возможность выполнить обучение сверточной нейронной сети. При положительном выборе, ИНС начнет обучаться до тех пор, пока ошибка обучения не достигнет 5%, в таком случае сеть считается обученной. Если же ошибка больше этого процента, то следует изменить параметры нейронной сети (пересоздать ИНС).

После обучения нейронная сеть сохраняется. Далее, пользователь имеет возможность открыть сверточную нейронную сеть заново (при условии, что

сеть сохранена ранее), а также тестировать искусственную нейросеть. Если при тестировании, результат распознавания верен, то сравнение происходит с имеющейся БД. При отрицательном результате – нейронную сеть необходимо переобучить.

Как описывалось ранее следующим этапом алгоритма будет – сравнение. Если результат и данные из БД совпадают, то выдается информация о ЛС, если же нет, в таком случае лекарственное средство либо просрочено, либо является фальсифицированным фармакологическим препаратом.

База данных(БД), используемая в данной программе, является собственностью аптечного пункта. В ней указана вся информация о ЛС и его изготовителе. БД была представлена на базе Microsoft SQL Server.[38][39]

Общий алгоритм программного средства представлен на рисунке 3.1.

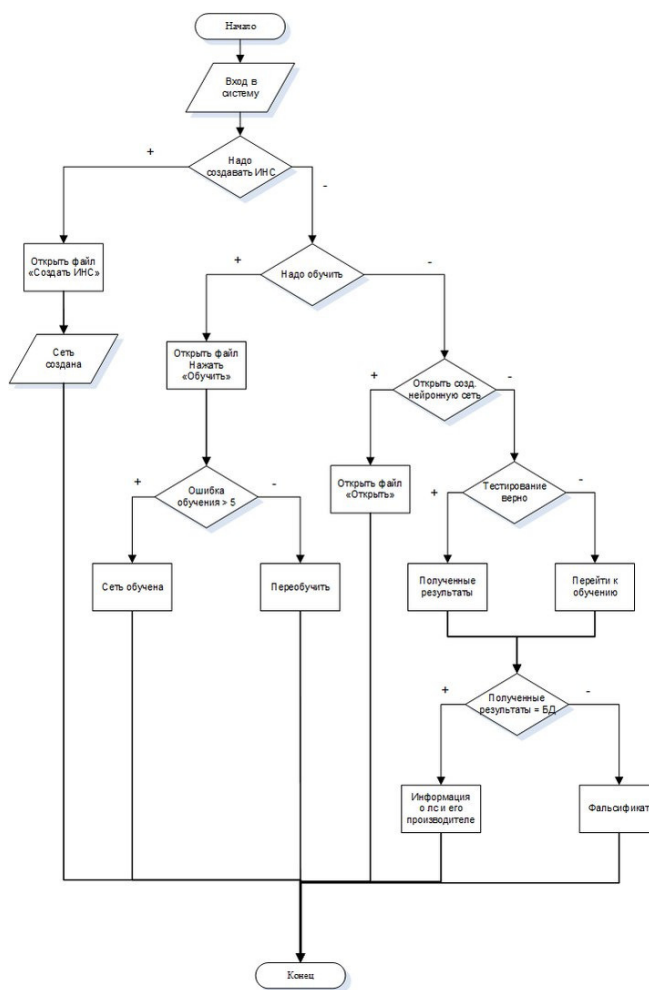


Рисунок 3.1 – Алгоритм программного обеспечения

3.4 Процесс обучения полученной нейросети

Для того чтобы обучить сеть необходимо подобрать идеальные параметры обучения: выбор количества итераций, выбор количества интервалов, скорость обучения и критерии обучения.

После, данные используются для расчета ранее описанного алгоритма обратного распространения ошибки.

Алгоритм обратного распространения:

1. Вычисляется производная ошибки по активности выходного элемента:

$$\frac{\partial E}{\partial Y_{jk}} = Y_{jk} - D_{jk} \quad (3.1)$$

2. Вычисляется производная ошибки по входу выходного элемента:

$$\frac{\partial E}{\partial X_{jk}} = \frac{\partial E}{\partial Y_{jk}} \cdot \frac{dY_{jk}}{dX_{jk}} = (Y_{jk} - D_{jk}) \cdot [Y_{jk} \cdot (1 - Y_{jk})] = \delta_{jk} \quad (3.2)$$

Здесь последний множитель (заклученный в квадратные скобки) является производной сигмоидальной функции

3. Вычисляется производная ошибки по весу связи:

$$\frac{\partial E}{\partial W_{il}^{jk}} = \frac{\partial E}{\partial X_{jk}} \cdot \frac{\partial X_{jk}}{\partial W_{il}^{jk}} = \delta_{jk} \cdot Y_{il} \quad (3.3)$$

4. Корректируется вес связи:

$$W_{il}^{jk}(t) = W_{il}^{jk}(t-1) + \Delta W_{il}^{jk} = W_{il}^{jk}(t-1) - \eta \frac{\partial E}{\partial W_{il}^{jk}}, \quad (3.4)$$

где η ($0 < \eta < 1$) - коэффициент скорости обучения.

5. Если дошли до 0-го слоя – выход.

6. Вычисляется (аналогично п. 1) производная ошибки по активности элемента предыдущего слоя:

$$\frac{\partial E}{\partial Y_{il}} = \sum_{k=1}^{N_j} \left(\frac{\partial E}{\partial X_{jk}} \cdot \frac{\partial X_{jk}}{\partial Y_{il}} \right) = \sum_{k=1}^{N_j} [\delta_{jk} \cdot W_{il}^{jk}] \quad (3.5)$$

Здесь суммирование производится по всем нейронам j -го слоя, связанным с данным нейроном i -го слоя.

7. Вычисляется (аналогично п. 2) производная ошибки по входу элемента i -го слоя:

$$\frac{\partial E}{\partial X_{il}} = \frac{\partial E}{\partial Y_{il}} \cdot \frac{dY_{il}}{dX_{il}} = \sum_{k=1}^{N_j} \delta_{jk} \cdot W_{il}^{jk} \cdot [Y_{il} \cdot (1 - Y_{il})] = \delta_{il} \quad (3.6)$$

8. Переход на п. 3 и продолжение вычислений с заменой индексов.

Емкостью НС называют число входных образов, которые она способна научиться распознавать. Для СНС детерминистская емкость сети C_d оценивается следующим образом:

$$\frac{L_w}{m} < C_d < \frac{L_w}{m} \log \left(\frac{L_w}{m} \right), \quad (3.7)$$

где L_w – число подстраиваемых весов, m – число нейронов в выходном слое.

В результате обучения и корректировки слоев нейронная сеть выглядит следующим образом.

3.5 Правила пользования разработкой конечным пользователем

Разработанное программное средство предназначается для программиста и конечного пользователя. У каждого из вышеперечисленного лица есть соответствующие роли и полномочия.

Роль «Программист» состоит из следующих полномочий:

- Вход в программное средство. Необходимо указать логин и пароль. Данная операция изображена на рисунке 3.2

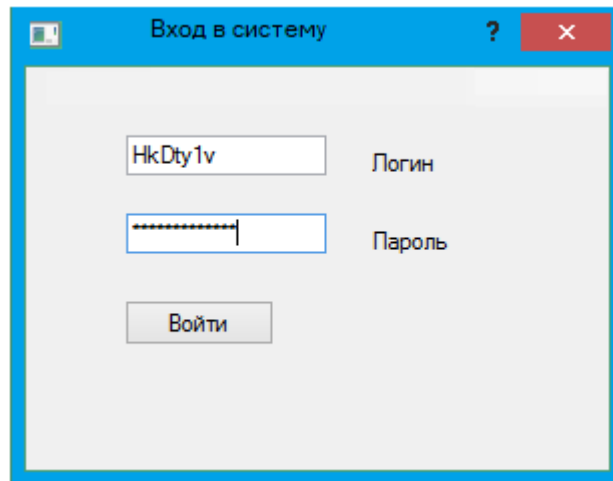


Рисунок 3.2 – Вход в систему

– Создание нейронной сети. Для формирования сверточной нейронной сети, необходимо открыть «Файл», далее нажать «Создать». Для реализации нейронного метода СНС, следует знать количество сверточных, субдискретизационных и нейросетевых слоев, а также нейросетевую архитектуру. Необходимо определить чередование слоев (выбор архитектуры). Форма, позволяющая создать сеть, представлена на рисунке 3.3

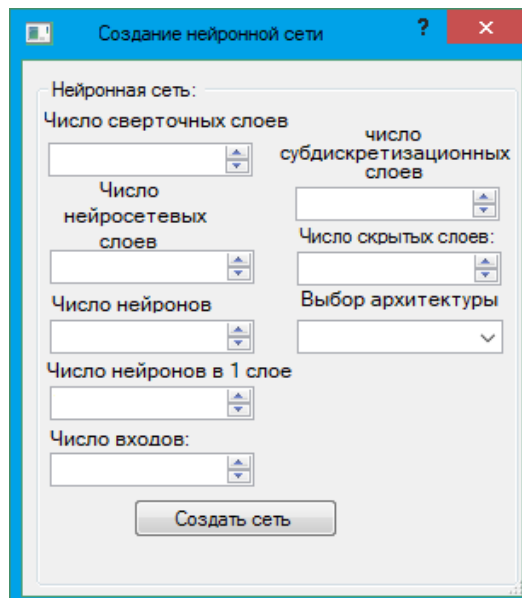


Рисунок 3.3 - Создание ИНС.

– Обучение нейронной сети. После того как сеть была создана ее необходимо обучить. Для данной операции следует выбрать папку, где находятся изображение, нажав на кнопку с тремя точками или ввести вручную в поле «Путь к обучающей выборке». Далее устанавливаем скорость обучения. Она не должна быть слишком высокой, но и не слишком медленной, так как от нее зависит качество обучения. Интервал скорости от 0,0001 до 0,2 единиц, при этом идеальной считается 0.1. Далее выбрать критерий остановки – процент ошибки при котором обучение закончится. Далее нажать «Start». Если, при обучении процент текущей ошибки не уменьшается и не равен заданному критерию остановки, то нажать кнопку «Stop». Форма обучения сверточной нейронной сети представлен на рисунке 3.4.

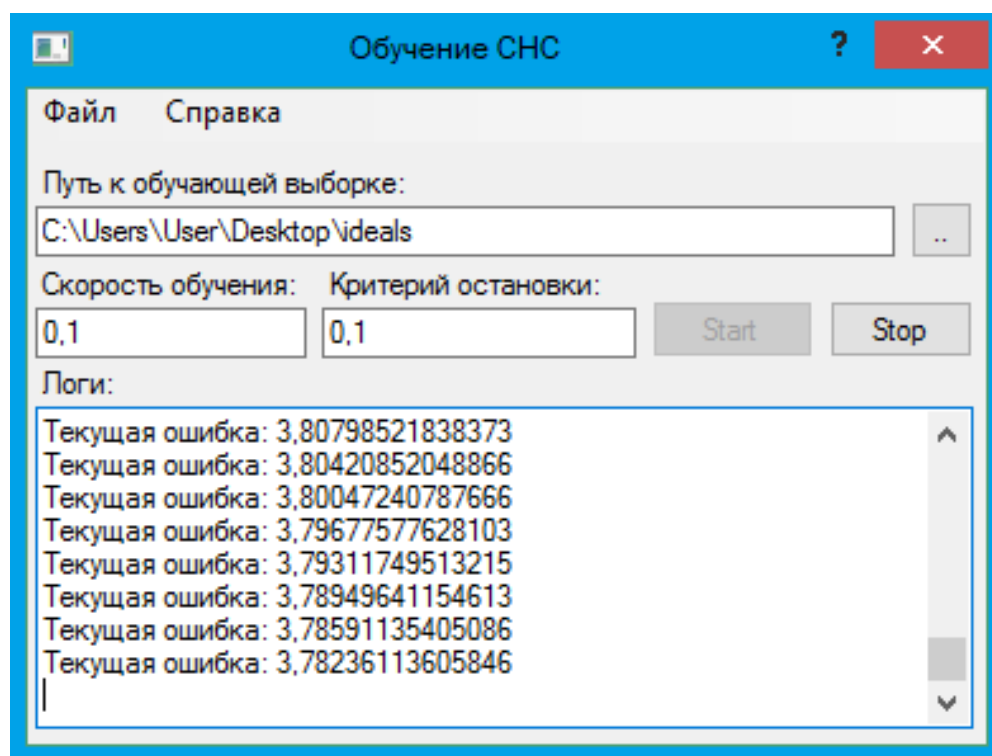


Рисунок 3.4 - Обучение НС.

– Сохранить полученные результаты – «Файл» нажать «Сохранить как...» выбрать путь и данные сохраняются вместе с обучением.

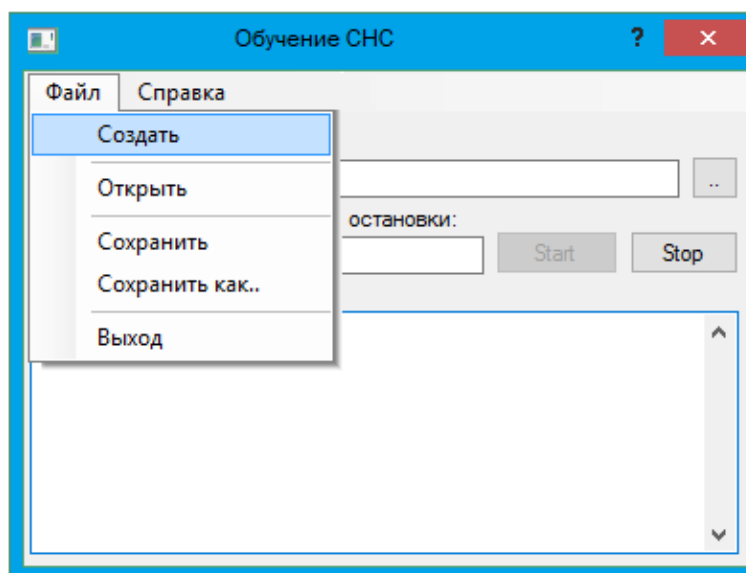


Рисунок 3.5 - Сохранение ИНС.

– Тестирование. Выбирать меню «Файл» - «Открыть». После этого откроется окно с выбором расположения тестовой выборки и вывода результата на экран (рисунок 3.6).

Далее нажать на кнопку «Проверить». Если изображение распознано, то результат отразится ниже.

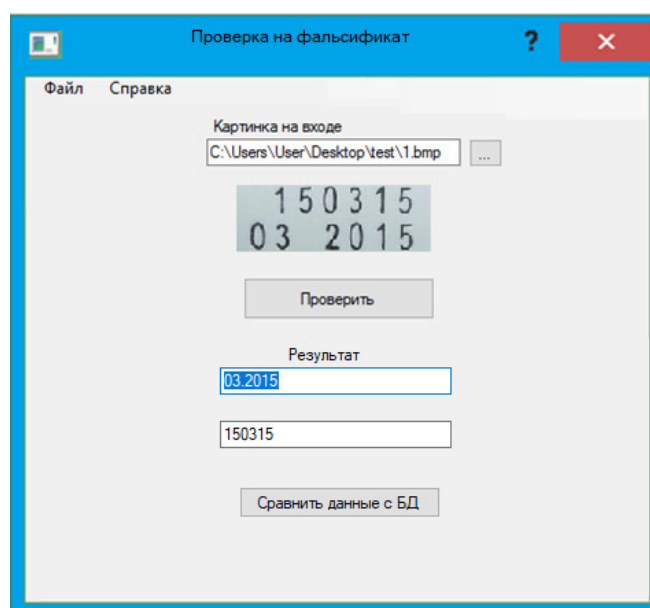
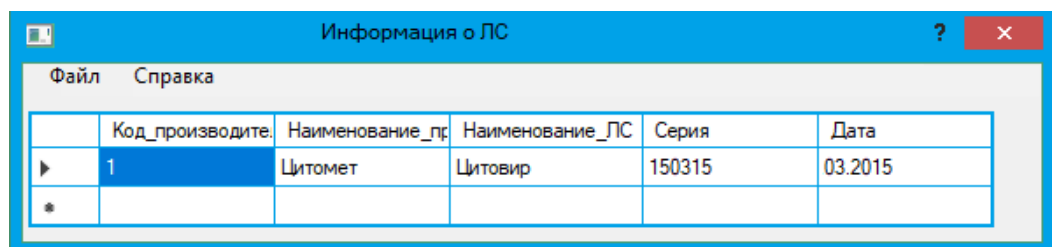


Рисунок 3.6 – Проверка на фальсификат (подлинный товар)

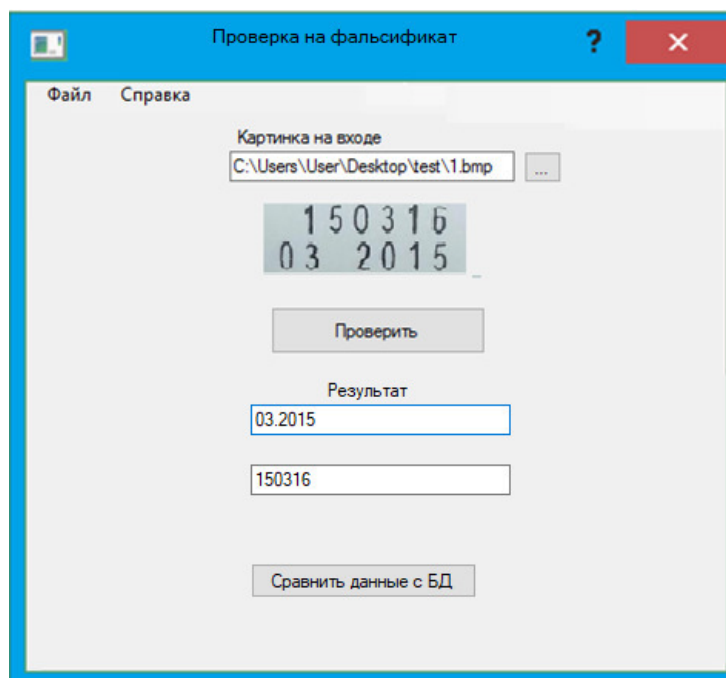
После нажатия на кнопку «сравнить данные с БД» можно наблюдать на рисунке 3.7



| | Код_производите. | Наименование_пр. | Наименование_ЛС | Серия | Дата |
|---|------------------|------------------|-----------------|--------|---------|
| ▶ | 1 | Цитомет | Цитовир | 150315 | 03.2015 |
| * | | | | | |

Рисунок 3.7 – информация о ЛС (подлинный товар)

Если серия и/или дата изготовления нету в БД, следовательно, товар считается не легальным – рисунок 3.8.



Проверка на фальсификат

Картинка на входе
C:\Users\User\Desktop\test\1.bmp

150316
03 2015

Проверить

Результат
03.2015

150316

Сравнить данные с БД

Рисунок 3.8 – Проверка на фальсификат (фальсифицированный товар)

Информация о фальсификации представлена на рисунках 3.9.

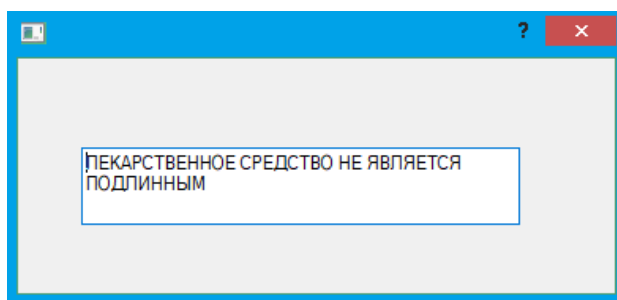


Рисунок 3.9 – Форма вывода сообщения о некачественном товаре

Если есть вопросы которых нет возможности решить, обращаться меню – «Справка» на всех формах программного средства.

Роль «Пользователей» состоит из следующих полномочий:

- Вход в систему.

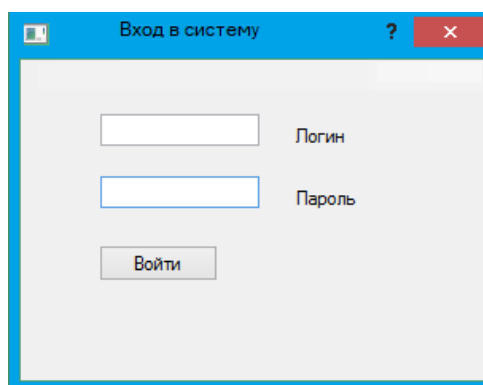


Рисунок 3.9 – Вход в систему для «Пользователя»

– Открыть сеть. На форме 3.10 представлено проверки на фальсификат. При нажатии на кнопку «Файл», затем «Открыть».

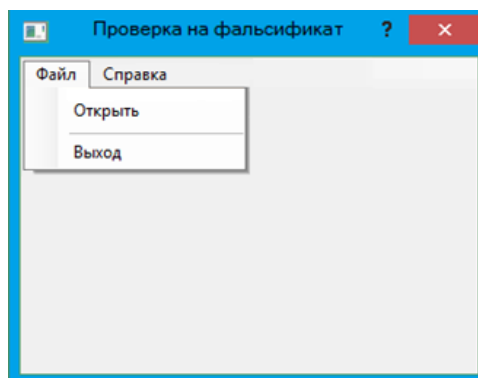


Рисунок 3.10 –Открыть сеть для «Пользователя»

После этого открывается форма, представленная на рисунке 3.11.

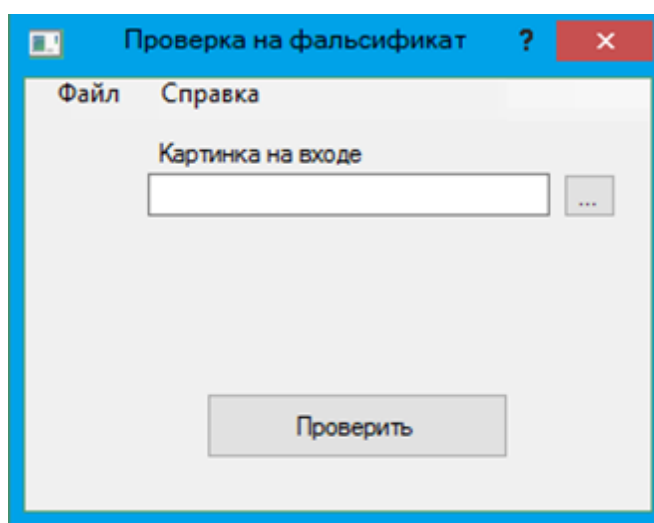


Рисунок 3.11 – Проверка на фальсификат для «Пользователя»

- Проверка на фальсификат. В данной операции выбрать фото для распознавания (рисунок 3.12)

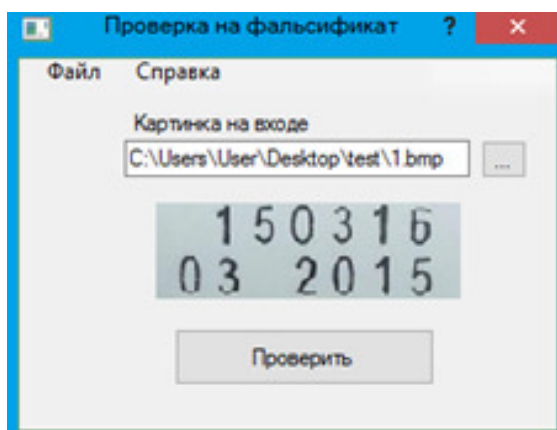


Рисунок 3.12 – Выбор фото для распознавания

Далее выводится результат. В данном случае это фальсификат. Пример представлен на рисунке 3.13.

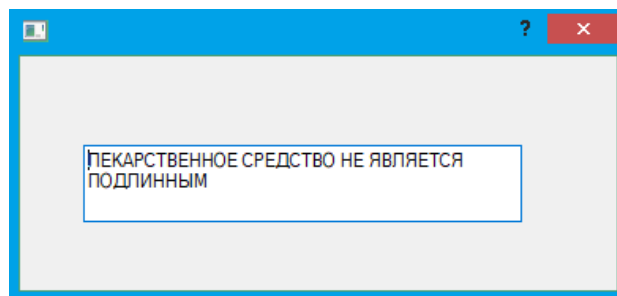


Рисунок 3.13 – Результат после проверки на фальсификат

Выводы по разделу

Для поставленной проблемы была реализована сверточная нейронная сеть, которая в последствии была обучена. Все весовые коэффициенты нейронов, количество слоев свертки и субдискретизации были подобраны экспериментально. Для разработки программного средства выбран язык Python и среда разработки PyQt за их эргономичность, скорость обработки данных и возможность работать с базой данных Microsoft SQL Server, а также за множество библиотек для глубинного машинного обучения. Библиотека Coffee эффективнее других подходит для реализации поставленной задачи. Так же были установлены правила пользования для пользователя и программиста.

4 Тестирование нейросетевой архитектуры

4.1 Характеристики исходных данных

В качестве исходных данных для проведения экспериментов по оценке эффективности предложенных подходов к построению модели наиболее эффективного классификатора были взяты данные даты и серии на упаковке товара. Образцы реальных данных были сформированы самостоятельно – это около 1000 фото(тестеров) предварительно отформатированных в графическом редакторе изображенных на рисунке 4.1.



Рисунок 4.1 – Пример идеальных тестеров

Тестеры приближены к идеалу, для того чтобы откорректировать весовые коэффициенты. После такого теста закончено успешно – изображения зашумляется и СНС опять проходит обработку при этом выполняется изменение коэффициентов, количество нейронов и возможно число слое в (сверточного и субдискретизации). Для дополнительного усложнения задачи классификации, 20 % имеющихся кадров были дополнительно зашумлены с помощью создания искусственных помех, эффекты от которых были максимально приближены к естественным помехам. Пример зашумленных тестеров представлен на рисунке 4.2 [36][18]

Присутствие естественных и добавление искусственных помех на кадры видео исходных данных позволяют с одной стороны усложнить задачу

классификации, и с другой стороны обучить сверточную нейронную сеть на как можно более широком наборе исходных данных, повышая её адаптивность к их изменению.



Рисунок 4.2 – Примеры искусственных помех на кадрах фото

Основываясь на наличии кадров фото и соответствующих им изображений-масок, образцы для экспериментов по обучению и тестированию классификаторов, основанных на сверточных нейронных сетях, были получены путем автоматического вырезания квадратных участков кадра, содержащего дату и серию на упаковках, и приведения их к единому размеру 12 x 18 пикселей. Принятие решения о размерах образцов было осуществлено на 62 основании размеров самых малых встречающихся на фото образцов цифр. Примеры образцов обучающей и тестовой выборки для экспериментов по классификации цифр, полученных с тестеров. Из них вручную были отобраны наиболее информативные образцы, примеры которых представлены на рисунке 4.3.



Рисунок 4.3 – Примеры образцов обучающей и тестовой выборки

С использованием полученных образцов, для проведения экспериментов, случайным образом были сформированы три обучающие выборки, состоящие 1000. Выборка состояла из неповторяющихся в других обучающих наборах образцов. Также была сформирована одна тестовая выборка, состоящая из 800 образцов, не присутствующих в обучающих выборках. Тестовая выборка также состояла из образцов, содержащих дату и серию, и образцов со случайными элементами, которые находились в равном отношении по их количеству в этом наборе данных. Именно с использованием данных выборок были произведены эксперименты по оценке эффективности предложенных 63 подходов к формированию наиболее эффективной архитектуры классификатора патологий, основанного на сверточных нейронных сетях.[6]

4.3 Результаты эксперимента.

Эксперименты по оценке эффективности предложенных подходов к формированию архитектуры классификатора на основе сверточных нейронных сетей предполагают получение параметров архитектуры СНС сравниваемых моделей с помощью разработанных алгоритмов, обучение и тестирование моделей наборах данных, обработку результатов тестирования с помощью построения кривой Receiver Operating Characteristic (ROC-кривая), а также сравнение и интерпретацию полученных результатов.

Для оценки результатов тестирования был проведен ROC-анализ, представляющий собой построение ROC-кривой, демонстрирующей взаимную зависимость доли верно классифицированных как тестеров (TPR или чувствительность классификатора) и доли неверно классифицированных случайных структур (FPR или 1 минус специфичность классификатора). На рисунках 4.4 представлен результат ROC-анализа

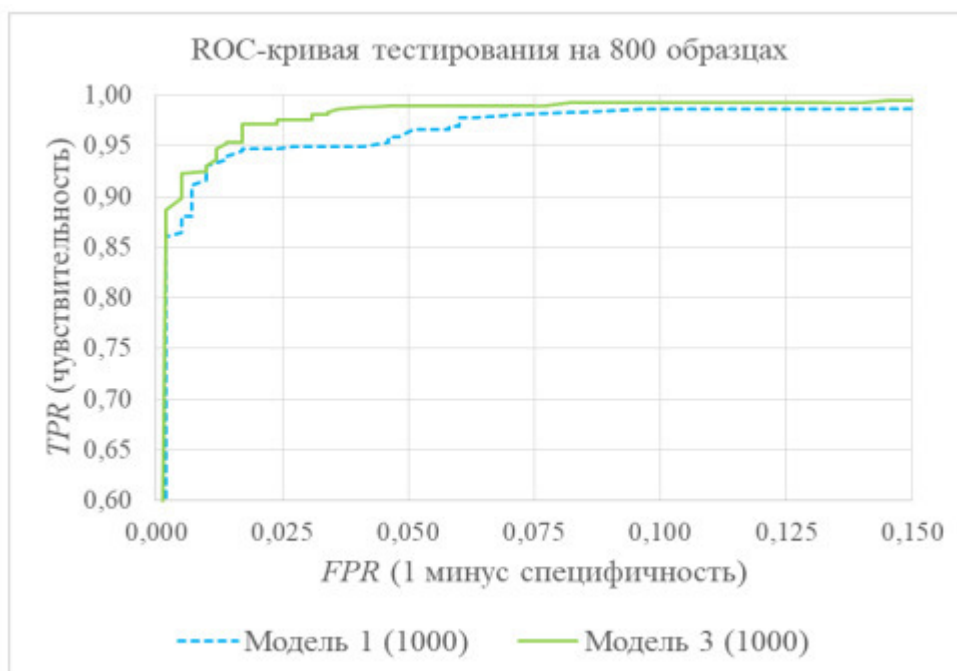


Рисунок 4.4 – ROC-кривая тестирования,

Как и при любом обучении есть ошибки, которые представлены на рисунке 4.5



Рисунок 4.5 – Примеры ошибочной выборки

Приведенные данные на рисунке 4.4-4.5 означают, что модель СНС будет верно классифицировать дату и серию на упаковках товара на каждом кадре фото с вероятностью 97 % при максимально возможных приблизительно семи ложных срабатываниях.

ЗАКЛЮЧЕНИЕ

В результате выполнения заданий магистерской диссертации исследована проблема фальсификата лекарственных средств на Российском фармакологическом рынке. Целью данной работы разработка интеллектуального метода, основанного на искусственной нейронной сети для распознавания текстовой информации с упаковок товаров, для выявления, просроченного и фальсифицированного лекарственного средства.

В ходе магистерского исследования решены следующие задачи:

- Анализ методов распознавания образов;
- Исследование архитектуры нейросети;
- Проектирование и разработка нейросети.
- Обучение выбранной искусственной нейросети;
- Тестирование нейросети.

На основании поставленных задач и требований был разработан интеллектуальный метод, на основе искусственной нейронной сети, архитектура которого – сверточные нейронные сети. Так как, для реализации проблемы она подходит лучше, чем многослойный персептрон и радиально-базисная сеть. СНС состоит из входного слоя двух сверточных, 1 субдифференцированного и 2х нейросетевых слоев. Такая архитектура показала себя наиболее подходящей для поставленной задачи.

Также, был выбран пороговый метод сегментации. В дальнейшем принято решение обучить нейронную сеть методом обратного распространения ошибки. Обучалась нейронная сеть на 870 изображений. Сначала приближенных к идеалу фотографий, затем на реальных. Это позволило качественней обучить сверточную нейронную сеть.

На основании вышеперечисленного был разработан алгоритм и программное средство, представленное с экранными формами. Также проведено тестирование на обучающей выборке, состоящей из 1000 тестерах,

которые были сформированы самостоятельно в процессе исследования магистерской диссертации

Как показала практика, реализованная искусственная нейронная сеть сумела распознать не все представленные тестеры. В основном это изображения низкого качества, где даже человеческий глаз не смог с точностью определить информацию на изображении. После тестирования был построен график, где четко показана ошибка распознавания, которая составила 3%. С уверенностью можно сказать, что реализованный интеллектуальный метод в полной мере решает поставленные задачи, а, следовательно, и цель.

В процессе исследования магистерской диссертации, была опубликована статья на тему График отгрузки стан 700 в издательстве «ВЕСТНИК СНО НИУ «БЕЛГУ»-2016», ВЫПУСК XX» в сборнике.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бельских, И. Е. Особенности регулирования цен на фармацевтическом рынке / И. Е. Бельских, С. В. Кулагина // Международный бухгалтерский учет. — 2011. — № 14. — С. 37–45.
2. Евстратов, А. В. Основные субъекты фармацевтического рынка Российской Федерации и их роль в повышении эффективности его функционирования / А. В. Евстратов, В. С. Игнатьева // Вестник Самарского государственного университета. — 2015. — № 9 (131). — С. 94–100.
3. Евстратов, А. В. Структурные параметры фармацевтического рынка Российской Федерации в 2001–2014 годах: закономерности формирования и тенденции развития / А. В. Евстратов // Известия Волгоградского государственного технического университета. — 2015. — № 3 (158). — С. 31–43.
4. Евстратов, А. В. Фармацевтический рынок Российской Федерации: основные структурные параметры развития в 2001–2013 годах / А. В. Евстратов // Современные технологии управления, 2015. — № 2 (50). — С. 27–36.
5. Форсайт Д.А., Понс Ж. Компьютерное зрение. Современный подход. - М.: Вильямс, 2012. – 928 с.
6. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - М.: Мир, 2014. – 184 с.
7. Хайкин С. Нейронные сети: полный курс, 2-е издание. – М.: Вильямс, 2012. – 1103 с.
8. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015. – 400 с.

9. Местецкий Л. М., «Математические методы распознавания образов», МГУ, ВМиК, Москва, 2011., с. 42 – 44
10. Андрейчиков, А.В. Интеллектуальные информационные системы [Текст]: учебник / А. В. Андрейчиков, О. Н. Андрейчикова. – М.: Финансы и статистика, 2004. – 424с.: ил.
11. Гаврилова, Т.А. Базы знаний интеллектуальных систем [Текст] / Т.А. Гаврилова, В. Ф. Хорошевский. – СПб: Питер, 2008. – 384 с.: ил.
12. Голубев М.Н., Апальков И.В. Устранение ложных срабатываний в алгоритме выделения лиц на изображении // Докл. 12-й междунар. конф. «Цифровая обработка сигналов и ее применение» (Б8РА-2010), Москва, 2010. Т.2. С. 302-304.
13. Трегубов А.А., Цопкало Н.Н. Контекстное нейросетевое распознавание символов с учетом словаря и переходных вероятностей. // Сборник трудов научно-практической конференции "Информационная безопасность", Россия, Таганрог, 28-31 мая 2002 г.
14. Спицын В.Г., Цой Ю.Р. Представление знаний в информационных системах: учебное пособие / Томский политехнический университет – Томск: Изд-во ТПУ, 2007. – 160 с.
15. Серль Дж. Разум человека компьютерная программа/ Дж. Серль // В мире науки. - 2005. - № 3 - С. 7-13.
16. Черчленд П. Может ли машина мыслить? / П. Черчленд, П.С.Черчленд // В мире науки. 2009. - № 3. - С. 14-21.
17. Бирман Э.Г. Сравнительный анализ методов прогнозирования //НТИ. Сер.2 2010. - № 1. - С. 11-16
18. Бусленко Н. П. Моделирование сложных систем / Н.П. Бусленко. - М.: Наука, 2011.-400 с.
19. Вахитов А.Т. Обзор алгоритмов стереозрения/ А.Т.Вахитов, Л.С.Гуревич, Д.В.Павленко. Стохастическая оптимизация в информатике, вып.4 / Под ред. О.Н.Граничина- СПб. : Из-во Санкт-Петербургского ун-та, 2008. 299 с.

20. Вапник В.Н. Восстановление зависимостей по эмпирическим данным./В.Н.Вапник М.: Наука, 2010. - 448с.
21. Вапник В.Н. Теория распознавания образов (Статистические проблемы обучения). М.: Наука, 2012. 416 с.
22. Вентцель Е. С. Исследование операций: задачи, принципы, методология / Е.С. Вентцель. М.: Наука, 2011. - 206 с.
23. Вилкас Э. Й. Оптимальность в играх и решениях / Э.Й. Вилкас. М.: Наука, 2012. - 256 с.
24. Вороновский Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. -Харьков: Основа, 2013. -112 с.
25. Галушкин А.И. Нейронные сети: основы теории. / А.И. Галушкин -М.: Горячая линия Телеком, 2010г. - 496 с.
26. Гилл Ф. Практическая оптимизация./Ф. Гилл, У.Мюррей, М. Райт -М.: Мир, 2013.
27. Гладков Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М.Курейчик/Под ред. В.М.Курейчика.-2-е изд., испр. И доп.-М. :ФИЗМАТЛИТ, 2006.-320с.
28. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 126 с.
29. Горбань А.Н. Нейрокомпьютер или аналоговый ренессанс/А.Н.Горбань // МИР ПК, 2010, № 10.
30. Горбань А.Н. Обучение нейронных сетей / А.Н.Горбань М.: СП Параграф, 20.-256 с.
31. Горелик А.А., Скрипкин В.А. Методы распознавания: учебное пособие для вузов. 3-е изд., перераб. и доп. М.: Высш. шк., 1989. -232 с.
32. Гринт З. MySQL. Карманный справочник. :Пер. с англ. / З.Гринт, К.Ньюман. М.: ООО «И.Д. Вильяме», 2006. - 224 с.

33. Кальченко Д.А. Нейронные сети на пороге будущего. / КомпьютПресс, 2005.
34. Каширина И. Л. Нейросетевые технологии: Учебно-методическое пособие для вузов / Изд-во ВГУ. 2008. 72 с
35. Круглов, В.В. Искусственные нейронные сети. Теория и практика: моногр. / В.В. Круглов, В.В. Борисов. - М.: Горячая линия - Телеком; Издание 2-е, стер., 2002. - 382 с.
36. Толкачев, С. Нейронное программирование диалоговых систем / С. Толкачев. - Москва: РГГУ, 2016. - 192 с.
37. Шестаков К.М. Теория принятия решений и распознавание образов : курс лекций / К.М. Шестаков. Мн. : БГУ, 2005. - 184 с.
38. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб.: Питер, 2013. - 240 с.
39. Кириллов, В.В. Введение в реляционные базы данных. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. - СПб.: БХВ-Петербург, 2012. - 464 с.
40. Вежневек В. Выделение связных областей в цветных и полутоновых изображениях [Электронный ресурс] / И.М. Журавель. – Режим доступа:
http://www.ict.edu.ru/lib/index.php?a=elib&c=getForm&r=resDesc&d=light&id_res=2397
41. Миронов И.С., Скурлаев С.В. Распознавание образов при помощи нейронной сети [Электронный ресурс]. / И.М. Журавель. – Режим доступа:
http://confonline.susu.ru/index.php?option=com_content&view=article&id=57:2011-05-06-04-36-21&catid=16:-2----&Itemid=18

ПРИЛОЖЕНИЕ А

```
from Tkinter import *
    from Common71 import *      #<<1>>
    from GUICommon71 import *  #<<2>>

class LED(GUICommon):         #<<3>>
    def __init__(self, master=None, width=25, height=25,
        appearance=FLAT,
        status=STATUS_ON, bd=1,
        bg=None,
        shape=SQUARE, outline="",
        blink=0, blinkrate=1,
        orient=POINT_UP,
            takefocus=0):
    # preserve attributes
    self.master = master
    self.shape = shape
    self.Colors = [None, Color.OFF, Color.ON, #<<4>>
        Color.WARN, Color.ALARM, '#00ffdd']
    self.status = status
    self.blink = blink
    self.blinkrate = int(blinkrate)
    self.on = 0
    self.onState = None
    if not bg:
        bg = Color.PANEL
    self.frame=Frame(master, relief=appearance, bg=bg, bd=bd,
        takefocus=takefocus)
    basesize = width
    d = center = int(basesize/2)
    if self.shape == SQUARE:
        self.canvas=Canvas(self.frame, height=height, width=width,
            bg=bg, bd=0, highlightthickness=0)
        self.light=self.canvas.create_rectangle(0, 0, width, height,
            fill=Color.ON)
    elif self.shape == ROUND:
        r = int((basesize-2)/2)
        self.canvas=Canvas(self.frame, width=width, height=width,
            highlightthickness=0, bg=bg, bd=0)
        if bd > 0:
            self.border=self.canvas.create_oval(center-r, center-r,
                center+r, center+r)
            r = r - bd
            self.light=self.canvas.create_oval(center-r-1, center-r-1,
                center+r, center+r,
                fill=Color.ON,
                outline=outline)
        else: # Default is an ARROW
            self.canvas=Canvas(self.frame, width=width, height=width,
                highlightthickness=0, bg=bg, bd=0)
            x = d
            y = d
    # <<5>>
    VL = ARROW_HEAD_VERTICES[orient] # Get the vertices for the arrow
    self.light=self.canvas.create_polygon(eval(VL[0]),
        eval(VL[1]), eval(VL[2]), eval(VL[3]),
        eval(VL[4]), eval(VL[5]), eval(VL[6]),
        eval(VL[7]), outline = outline)
```

```

self.canvas.pack(side=TOP, fill=X, expand=NO)
self.update()
def update(self):
# First do the blink, if set to blink
if self.blink:
if self.on:
if not self.onState:
self.onState = self.status
self.status = STATUS_OFF
self.on = 0
else:
if self.onState:
self.status = self.onState # Current ON color
self.on = 1
# Set color for current status
self.canvas.itemconfig(self.light, fill=self.Colors[self.status])
self.canvas.update_idletasks()
if self.blink:
self.frame.after(self.blinkrate * 1000, self.update)
if __name__ == '__main__':
class TestLEDs(Frame):
def __init__(self, parent=None):
# List of Colors and Blink On/Off
states = [(STATUS_OFF, 0),
(STATUS_ON, 0),
(STATUS_WARN, 0),
(STATUS_ALARM, 0),
(STATUS_SET, 0),
(STATUS_ON, 1),
(STATUS_WARN, 1),
(STATUS_ALARM, 1),
(STATUS_SET, 1)]
# List of LED types to display,
# with sizes and other attributes
leds = [(ROUND, 15, 15, FLAT, 0, None, "gray50"),
(ROUND, 15, 15, RAISED, 1, None, ""),
(SQUARE, 12, 12, SUNKEN, 1, None, ""),
(SQUARE, 8, 8, FLAT, 0, None, ""),
(SQUARE, 8, 8, RAISED, 0, None, ""),
(SQUARE, 16, 8, FLAT, 1, None, ""),
(ARROW, 14, 14, RIDGE, 1, POINT_UP, ""),
(ARROW, 14, 14, RIDGE, 0, POINT_LEFT, "black"),
(ARROW, 14, 14, FLAT, 0, POINT_DOWN, "yellow")]
Frame.__init__(self) # Do superclass init
self.pack()
self.master.title('LED - Stage 2') # Label Toplevel shell
for shape, w, h, app, bd, orient, outline in leds:
frame = Frame(self, bg=Color.PANEL)
frame.pack(anchor=N, expand=YES, fill=X)
# Iterate for selectes states
for state, blink in states:
LED(frame, shape=shape, status=state,
width=w, height=h, appearance=app,
orient=orient, blink=blink, bd=bd,
outline=outline).frame.pack(side=LEFT,
expand=YES,
padx=1, pady=1)
TestLEDs().mainloop()
import re
for test_string in ['555-1212', 'ILL-EGAL']:
if re.match(r'^\d{3}-\d{4}$', test_string):
print test_string, 'is a valid US local phone number'

```

```

else:
    print test_string, 'rejected'
prices = {'neiro': 0.40, 'koeff': 0.50}
my_purchase = {
    'apple': 1,
    'banana': 6}
grocery_bill = sum(prices[fruit] * my_purchase[fruit]
                   for fruit in my_purchase)
print 'I owe the grocer $%.2f' % grocery_bill
# This program adds up integers in the command line
import sys
try:
    total = sum(int(arg) for arg in sys.argv[1:])
    print 'sum =', total
except ValueError:
    print 'Please supply integer arguments'
# indent your Python code to put into an email
import glob
# glob supports Unix style pathname extensions
python_files = glob.glob('*.py')
for file_name in sorted(python_files):
    print '  -----' + file_name

    with open(file_name) as f:
        for line in f:
            print '    ' + line.rstrip()

    print

activities = {8: 'Sleeping',
             9: 'Commuting',
             17: 'Working',
             18: 'Commuting',
             20: 'Eating',
             22: 'Resting' }

time_now = localtime()
hour = time_now.tm_hour

for activity_time in sorted(activities.keys()):
    if hour < activity_time:
        print activities[activity_time]
        break
else:
    print 'Unknown, AFK or sleeping!'
REFRAIN = """
%d bottles of beer on the wall,
%d bottles of beer,
take one down, pass it around,
%d bottles of beer on the wall!
"""
bottles_of_beer = 99
while bottles_of_beer > 1:
    print REFRAIN % (bottles_of_beer, bottles_of_beer,
                    bottles_of_beer - 1)
    bottles_of_beer -= 1
class BankAccount(object):
    def __init__(self, initial_balance=0):
        self.balance = initial_balance
    def deposit(self, amount):
        self.balance += amount
    def withdraw(self, amount):

```

```

        self.balance -= amount
    def overdrawn(self):
        return self.balance < 0
my_account = BankAccount(15)
my_account.withdraw(5)
print my_account.balance
def median(pool):
    copy = sorted(pool)
    size = len(copy)
    if size % 2 == 1:
        return copy[(size - 1) / 2]
    else:
        return (copy[size/2 - 1] + copy[size/2]) / 2
class TestMedian(unittest.TestCase):
    def testMedian(self):
        self.failUnlessEqual(median([2, 9, 9, 7, 9, 2, 4, 5, 8]), 7)
if __name__ == '__main__':
    unittest.main()
def median(pool):
    """Statistical median to demonstrate doctest.
    >>> median([2, 9, 9, 7, 9, 2, 4, 5, 8])
    7
    """
    copy = sorted(pool)
    size = len(copy)
    if size % 2 == 1:
        return copy[(size - 1) / 2]
    else:
        return (copy[size/2 - 1] + copy[size/2]) / 2
if __name__ == '__main__':
    import doctest
    doctest.testmod()
lines = ""
This is the
first paragraph.

This is the second.
"".splitlines()
# Use itertools.groupby and bool to return groups of
# consecutive lines that either have content or don't.
for has_chars, frags in groupby(lines, bool):
    if has_chars:
        print ''.join(frags)
# PRINTS:
# This is the first paragraph.
# This is the second.

# write stocks data as comma-separated values
writer = csv.writer(open('stocks.csv', 'wb', buffering=0))
writer.writerows([
    ('GOOG', 'Google, Inc.', 505.24, 0.47, 0.09),
    ('YHOO', 'Yahoo! Inc.', 27.38, 0.33, 1.22),
    ('CNET', 'CNET Networks, Inc.', 8.62, -0.13, -1.49)
])

# read stocks data, print status messages
stocks = csv.reader(open('stocks.csv', 'rb'))
status_labels = {-1: 'down', 0: 'unchanged', 1: 'up'}
for ticker, name, price, change, pct in stocks:
    status = status_labels[cmp(float(change), 0.0)]
    print '%s is %s (%s%%)' % (name, status, pct)
BOARD_SIZE = 8

```



```

def under_attack(col, queens):
    left = right = col

    for r, c in reversed(queens):
        left, right = left - 1, right + 1

        if c in (left, col, right):
            return True
    return False

def solve(n):
    if n == 0:
        return [[]]

    smaller_solutions = solve(n - 1)

    return [solution+[(n,i+1)]
            for i in xrange(BOARD_SIZE)
            for solution in smaller_solutions
            if not under_attack(i+1, solution)]
for answer in solve(BOARD_SIZE):
    print answer
def iter_primes():
    # an iterator of all numbers between 2 and +infinity
    numbers = itertools.count(2)

    # generate primes forever
    while True:
        # get the first number from the iterator (always a prime)
        prime = numbers.next()
        yield prime

    # this code iteratively builds up a chain of
    # filters...slightly tricky, but ponder it a bit
    numbers = itertools.ifilter(prime.__rmod__, numbers)

for p in iter_primes():
    if p > 1000:
        break
    print p

pantry = set(['olive oil', 'pesto'])
for ingredient in tree.getiterator('tr'):
    amt, unit, item = ingredient
    if item.tag == "td" and item.text not in pantry:
        print "%s: %s %s" % (item.text, amt.text, unit.text)
BOARD_SIZE = 8

class BailOut(Exception):
    pass

def validate(queens):
    left = right = col = queens[-1]
    for r in reversed(queens[:-1]):
        left, right = left-1, right+1
        if r in (left, col, right):
            raise BailOut

def add_queen(queens):
    for i in range(BOARD_SIZE):
        test_queens = queens + [i]

```

```

try:
    validate(test_queens)
    if len(test_queens) == BOARD_SIZE:
        return test_queens
    else:
        return add_queen(test_queens)
except BailOut:
    pass
raise BailOut

queens = add_queen([])
print queens
print "\n".join(" " * q + "Q " + " " * (BOARD_SIZE - q - 1) for q in queens)

guesses_made = 0

name = raw_input('Hello! What is your name?\n')

number = random.randint(1, 20)
print 'Well, {0}, I am thinking of a number between 1 and 20.'.format(name)

while guesses_made < 6:

    guess = int(raw_input('Take a guess: '))

    guesses_made += 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    print 'Good job, {0}! You guessed my number in {1} guesses!'.format(name, guesses_made)
else:

```


Магистерская диссертация выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« » 201 г.

(подпись)

Беспалова А.В.

(ФИО)