Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

ПРОЕКТ ПРОВЕРЕН
Рецензент, к.т.н., доцент,
_____(Я.С. Добрынина)
«____» _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, д.т.н.,с.н.с.
_____ (Б.М. Суховилов)
«____» _____ 2019 г.

Разработка математических моделей для распознавания эмоций человека с помощью сверточных нейронных сетей

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
ЮУрГУ–38.04.05.62.2019.130.ПЗ ВКР

Руководитель проекта, д.т.н., профессор
_____ (В.В. Мокеев)
«____» _____ 2019 г.

Автор проекта,
студент группы ЭУ– 244
_____ (А.С. Тёмкин)
«____» _____ 2019 г.

Нормоконтролер, к.т.н., доцент
_____ (Е.В. Бунова)
«____» _____ 2019 г.

Челябинск 2019

АННОТАЦИЯ

Тёмкин А.С. Разработка математических моделей для распознавания эмоций человека с помощью сверточных нейронных сетей
– Челябинск: ЮУрГУ, ЭУ–244, 85 с., 33 ил., 5 табл., библиогр. список – 33 наим., прил. – 1.

Магистерская работа посвящена исследованию математических моделей для распознавания эмоций человека с помощью сверточных нейронных сетей.

В квалификационной работе рассматривается понятие сверточной нейронной сети и машинного зрения, производится сравнения методов, строятся модели по выбранным методам, разрабатывается план коммерциализации проекта.

Основные задачи работы:

1) определение понятия компьютерного зрения;
2) теоретическое описание машинного обучения, его виды и алгоритмы;
3) анализ и сравнение имеющихся методов машинного обучения;
4) формулировка требований к точности распознавания;
5) построение математической модели;
6) сравнительный анализ результатов точности распознавания;
7) формирование плана коммерциализации.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы обусловлена тем, что в настоящее время проблема современного мира заключается в быстром развитии компьютерных технологий и влияния на подрастающее поколение. Дети сегодня начинают использовать различные электронные устройства еще раньше нежели начинают говорить. Отсюда то и появляется проблема актуальности данной работы построения математических моделей для распознавания эмоций человека с помощью сверточных нейронных сетей.

Отслеживать постоянно контент, который смотри ребенок практически невозможно. Отсюда следует, что в этом случае отслеживания эмоций ребенка поможет понять какой спектр эмоций он испытывает. Например, гнев, отвращение, страх, счастье, грусть, удивление или же спокойное отношение. Современные мультфильмы, которые подходят по возвратному рейтингу для детей, могут нанести вред психике ребенка, вызвать страх или гнев.

Магистерская диссертация выполнена с целью разработки математических моделей для распознавания эмоций человека с помощью сверточных нейронных сетей. Анализ методов выявил, что наилучшим подходом может представлять собой использование сверточной нейронной сети.

Статистические исследования машинного обучения дали большой набор алгоритмических и математических инструменты за последние десятилетия, и породил ряд коммерческих и научных приложений. Тем не менее, некоторые из первоначальных целей этой области исследований остаются недостижимы. Цель исследований в области машинного обучения заключается в создании методов что позволит компьютерному зрению, способным к сложному обучению с минимальным вмешательством человека и предварительными знаниями распознавать сложно отличимые объекты на изображении или видео ряде.

Научная новизна состоит в том, что были рассмотрены и проанализированы методы машинного обучения, проведен сравнительный анализ методов, построены и проанализированы модели сверхточных нейронных сетей.

В качестве объекта исследования выступают методы распознавания объектов по их изображениям.

Предметом является процесс построения и обучения сверточной нейронной сети.

Практическая значимость. Разработанная модель свёрточной нейронной сети и алгоритм её обучения послужили основой для создания прототипа программы распознавания эмоций детей. Разработанные в диссертации модели распознавания предназначаются для использования в системах контроля психического состояния ребенка.

В работе проанализированы теоретико-методологические основы распознавания эмоций по их изображениям. Разработана модель, реализующая распознавание эмоций и анализирующая их эффективность. Проведено исследование моделей распознавания и обоснован выбор наиболее эффективной.

Методологической основой исследования в магистерской диссертации послужили системный подход и диалектический метод научного познания. В ходе исследования мы использовали такие общенаучные приемы и методы как анализ исследуемых явлений и результатов, индукция, дедукция и формализация, научная абстракция, методы детерминированного факторного анализа, синтеза, методы группировки. Также применены диалектический, сравнительно-правовой, логический и системно-структурный методы исследования.

Целью исследования является анализ эффективной модели для распознавания эмоций по их изображениям.

Для достижения поставленной цели были выполнены следующие задачи:

- исследованы современные алгоритмы и методы распознавания эмоций;
- разработана и проанализирована наилучшая модель распознавания эмоций человека по изображениям;
- проведена оценка точности распознавания в сравнении с другими проанализированными методами.

Анализ научных работ сверточных нейронных сетей показывает, что ученые уже более 50-ти лет с различных методологических позиций обращались к проблемам в распознавании лиц и эмоций. Существование направлений машинного обучения рассматривается в работах отечественных и зарубежных ученых.

1 ТЕОРИТИЧЕСКИЕ ОСНОВЫ МЕТОДОВ, ТЕХНОЛОГИЙ И АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

1.1 Методологические основы машинного обучения

Распознавание лиц существует уже десятки лет, но резко эта технология начала развивается только в последние годы благодаря достижениям в областях компьютерного зрения и искусственного интеллекта. Сейчас эта технология используется для распознания людей на границе, разблокировки телефонов, поимки преступников и подтверждения банковских операций. Но некоторые технологические компании говорят, что алгоритмы распознавания лиц могут анализировать и наше эмоциональное состояние. С 1970 года психологи способны распознавать скрытые эмоции, изучая «микровыражения» на фотографиях и видео. Алгоритмы и камеры высокой четкости справятся с этим даже лучше и быстрее, говорят в технологических компаниях. Супермаркеты могут использовать эту технологию в проходах, не для установления личности, а для анализа публики - в плане возраста, пола, а также их настроения. Это может помочь в целевом маркетинге и при размещении товаров. Это может показаться сомнительным, но некоторые стартапы предлагают «выявление эмоций» в качестве меры безопасности. Британская компания WeSee, утверждает, что искусственный интеллект выявляет подозрительное поведение, «читая» незаметные невооруженному глазу подсказки. Такие эмоции, как сомнение и злость, могут скрываться под маской и контрастировать с тем, что говорит человек. WeSee утверждает, что работает с правоохранительной организацией, анализируя интервьюируемых людей. В будущем видеокамеры на платформах станций метро смогут использовать эту технологию для распознания подозрительного поведения и оповещать власти о потенциальной террористической угрозе. Но тут появляется проблема точности когда нужно просто распознать лицо, все еще нередко происходят ошибки – даже лучшие компании говорят, что могут распознать людей с точностью 90–92 процента. Когда вы пытаетесь оценить еще и эмоции,

вероятность ошибки намного больше. В Австралии полиция использует систему автоматического распознавания лиц (AFR), которая подключена к камерам наблюдения, и программное обеспечение от компании NEC, чтобы идентифицировать «искомых людей» и сравнивать их лица с базой данных по лицам в розыске. Ярким примером работоспособности системы стало то, что людей, которые находятся в розыске находят в течение очень короткого промежутка с момента, как программа была установлена, однако, системе все же не хватает точности распознавания, это говорит то, что технология дала множество «ложных» результатов на других [1].

Машинное обучение – не только математическая, но и практическая, инженерная дисциплина. Чистая теория, как правило, не приводит сразу к методам и алгоритмам, применимым на практике. Чтобы заставить их хорошо работать, приходится изобретать дополнительные эвристики, компенсирующие несоответствие сделанных в теории предположений условиям реальных задач. Практически ни одно исследование в машинном обучении не обходится без эксперимента на модельных или реальных данных, подтверждающего практическую работоспособность метода [5].

Рассмотрим существующие методы машинного обучения:

– деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение;

– ансамбли (Стекинг, Беггинг, Бустинг) используются повсеместно, в связи с их высокой точностью. Идея ансамблей очень просто, достаточно взять несколько не очень эффективных методов обучения и обучить их так что бы они исправляли ошибки друг друга, при таком подходе качество будет намного выше нежели использовать каждый метод по отдельности;

– ансамбли деревьев решений (случайный лес деревьев решений и градиентный бустинг). Построение множества деревьев даст N деревьев с разной

степенью переобученности, и путем простого усреднения результата уменьшится переобучение;

– нейронные сети – это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования прямиком из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию.

Сегодня технологическое развитие сверхточных нейронных сетей – представляет расширенную форму машинного обучения, которая предполагает более высокий уровень точности. Эти алгоритмы позволяют компьютерам анализировать изображения под разными углами и в разных масштабах. Можно распознавать лица намного точнее, даже если они частично закрыты очками или шарфами. Коэффициент ошибок снизился на порядок.

Сейчас мы можем распознавать подозрительное поведение, но не намерение, чтобы предотвратить что-либо. Но, думаю, все идет к тому, и мы уже делаем тесты в этой области.

В данной работе сверхточную нейронную сеть мы можем применить для распознавания эмоций детей. Целью нейронной сети это безопасный контроль за электронными устройствами. При использовании каких-либо гаджетов детьми, родители могут не понимать, что они могут нанести вред психике ребенка, даже если он пользуется контентом с правильным возвратным рейтингом. Полностью отследить чем занимается ребенок можно с помощью детского аккаунта Google. Там можно увидеть сколько времени и во что ребенок играл на планшете, вести контроль за установленными приложениями, ограничивать по времени то или иное приложение, но все это не показывает реальную картину, все дети разные, кто-то любит одно, а кто-то другое. Дети способны найти такой видео или игровой контент, который их начнет злить, на протяжении определенного времени. И отследить это в реальном времени сложно если никто не приглядывает за ребенком. Целью данного приложения будет анализ эмоции ребенка при просмотре видео или игр, в реальном времени, и идея такая что если ребенок начинает проявлять

агрессию или гнев по отношению в данному контенту, то можно подавать сигнал родителям, приостанавливать приложение или блокировать экран устройства.

Существует большое количество различных методов детектирования лиц [1, 4, 8, 14, 20]. В настоящее время наилучшее качество показывают алгоритмы, основанные на применении сверточных нейронных сетей (Convolutional Neural Network, CNN). Различия алгоритмов данной группы, как правило, обнаруживаются в архитектурах сетей и параметрах их обучения [6, 9]. Наряду с этим, могут отличаться методы обработки признаков [4, 12], а также подходы к решению проблемы поиска лиц разного масштаба [7].

Множество методов, основанных на применении сверточных нейронных сетей, можно разделить на две группы:

− методы, использующие выход предварительно обученной сверточной сети в качестве признакового описания изображения [8]. Построенное признаковое описание далее используется для обучения традиционных классификаторов;

− методы, предоставляющие на выходе нейронной сети полную информацию о расположении лиц, областей или окаймляющих лица прямоугольников [9].

Рассмотрим более подробно четыре метода детектирования лиц, основанные на применении сверточных нейросетей и демонстрирующие одни из лучших результатов на широко известном наборе данных.

1.2 Сравнительный анализ подходов к машинному обучению

Проблему распознавания эмоций поднимают в своих совместной работе работе Эйман Канджо, Кафедры вычислительной техники и технологий, Университет Ноттингема Трента, Ноттингем, Великобритания. Эман М.Г. Юнис б, Чи Сян Анг. Факультет компьютеров и информации, Университет Миния, Миния, Египет. Школа инженерии и цифровых искусств, Университет Кента, Великобритания. В данной работе они говорят о том, что обнаружение эмоций часто требует моделирования различных входных данных от множества моделей, включая физиологические сигналы, данные об окружающей среде (например, аудио и

погода), видео (например, для захвата выражений лица и жестов) и, в последнее время, движения и данные о местоположении. Многие традиционные алгоритмы машинного обучения использовались для сбора разнообразных мультимодальных данных на датчиках и уровнях характеристик для классификации человеческих эмоций. Хотя процессы разработки функций, часто встроенные в эти алгоритмы, полезны для моделирования эмоций, они наследуют некоторые критические ограничения, которые могут препятствовать разработке надежных и точных моделей. В этой работе они применяют метод глубокого обучения для классификации эмоций посредством итеративного процесса, добавляя и удаляя большое количество сигналов датчиков из разных модальностей. Их набор данных был собран в реальном исследовании со смартфонов и носимых устройств.

Другой работе Бабер Дж., Бахтяр М., Уддин Ахмед К., Нур В., Деви В., Саммад А. из Департамента CS и IT, Университет Белуджистана, Кветта, Пакистан. Университета Гренобля Альпы, Гренобль, Франция и Факультета компьютерных наук, Университет Хабиба, Карачи, Пакистан. В этой статье говорится о производительности распознавания выражений лица с использованием глубокой сверточной нейронной сети (DCNN. Эмоции включают в себя гнев, отвращение, страх, счастье, грусть, удивление и нейтральность. Также проведена оценка точности и анализ ложных срабатываний. Все классификаторы дают низкие показатели по наборам данных Fer2013. DCNN дает 54,46 % точности в тесте и 89,52 % в тренировочном наборе, тогда как в случае различных ядер машин опорных векторов (SVM) самая высокая точность составляет 45 % при использовании кубического ядра в обучающем наборе. Эксперименты показывают, что определенные выражения лица имеют больше ложных срабатываний, и лишь немногие из них являются очень преобладающими. В случае выражения Отвращения, в качестве доминантного ложного срабатывания используется Агрессия. Основываясь на анализе ложных срабатываний, классификаторы могут быть обучены для повышения точности выражений с доминирующими ложными срабатываниями.

Из данной работы видно, что точность распознавания эмоций всего 54,4 %, что является очень маленьким значением.

1.3 Постановка задачи

Дано конечное множество прецедентов (объектов, ситуаций), по каждому из которых собраны (измерены) некоторые данные. Данные о прецеденте называют также его описанием. Совокупность всех имеющихся описаний прецедентов называется обучающей выборкой. Требуется по этим частным данным выявить общие зависимости, закономерности, взаимосвязи, присущие не только этой конкретной выборке, но вообще всем прецедентам, в том числе тем, которые ещё не наблюдались. Говорят, также о восстановлении зависимостей по эмпирическим данным – этот термин был введён в работах Вапника и Червоненкиса [9].

Наиболее распространённым способом описания прецедентов является признаковое описание. Фиксируется совокупность n показателей, измеряемых у всех прецедентов. Если все n показателей числовые, то признаковые описания представляют собой числовые векторы размерности n. Возможны и более сложные случаи, когда прецеденты описываются временными рядами или сигналами, изображениями, видеорядами, текстами, попарными отношениями сходства или интенсивности взаимодействия, и т. д.

Для решения задачи обучения по прецедентам в первую очередь фиксируется модель восстанавливаемой зависимости. Затем вводится функционал качества, значение которого показывает, насколько хорошо модель описывает наблюдаемые данные. Алгоритм обучения (learning algorithm) ищет такой набор параметров модели, при котором функционал качества на заданной обучающей выборке принимает оптимальное значение. Процесс настройки (fitting) модели по выборке данных в большинстве случаев сводится к применению численных методов оптимизации.

Замечание о терминологии. В зарубежных публикациях термин algorithm употребляется только в указанном выше смысле, то есть это вычислительная

процедура, которая по обучающей выборке производит настройку модели. Выходом алгоритма обучения является функция, аппроксимирующая неизвестную (восстанавливаемую) зависимость. В задачах классификации аппроксимирующую функцию принято называть классификатором (classifier), концептом (concept) или гипотезой (hypothesys); в задачах восстановления регрессии – функцией регрессии; иногда просто функцией. В русскоязычной литературе аппроксимирующую функцию также называют алгоритмом, подчёркивая, что и она должна допускать эффективную компьютерную реализацию.

Задача сокращения размерности (dimensionality reduction) заключается в том, чтобы по исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки. В классе линейных преобразований наиболее известным примером является метод главных компонент.

Задача заполнения пропущенных значений (missing values) – замена недостающих значений в матрице объекты–признаки их прогнозными значениями.

Частичное обучение (semi-supervised learning) занимает промежуточное положение между обучением с учителем и без учителя. Каждый прецедент представляет собой пару «объект, ответ», но ответы известны только на части прецедентов. Пример прикладной задачи – автоматическая рубрикация большого количества текстов при условии, что некоторые из них уже отнесены к каким-то рубрикам.

Динамическое обучение (online learning) может быть, как обучением с учителем, так и без учителя. Специфика в том, что прецеденты поступают потоком. Требуется немедленно принимать решение по каждому прецеденту и одновременно доучивать модель зависимости с учётом новых прецедентов. Как и в задачах прогнозирования, здесь существенную роль играет фактор времени.

Активное обучение (active learning) отличается тем, что обучаемый имеет возможность самостоятельно назначать следующий прецедент, который станет известен. См. также Планирование экспериментов.

Многозадачное обучение (multi-task learning). Набор взаимосвязанных или схожих задач обучения решается одновременно, с помощью различных алгоритмов обучения, имеющих схожее внутренне представление. Информация о сходстве задач между собой позволяет более эффективно совершенствовать алгоритм обучения и повышать качество решения основной задачи.

Индуктивный перенос (inductive transfer). Опыт решения отдельных частных задач обучения по прецедентам переносится на решение последующих частных задач обучения. Для формализации и сохранения этого опыта применяются реляционные или иерархические структуры представления знаний.

Иногда к метаобучению ошибочно относят построение алгоритмических композиций, в частности, бустинг; однако в композициях несколько алгоритмов решают одну и ту же задачу, тогда как метаобучение предполагает, что решается много разных задач.

1.4 Выводы по главе 1

Итак, машинное обучение это – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

По итогам анализа можно сказать, что на сегодняшний день точность распознавания эмоций не высокая. Необходимо достичь более точного распознавания путём тонкой настройки нейронной сети.

В настоящее время существует множество методов, технологий и алгоритмов машинного обучения, каждый из которых содержит свои достоинства и недостатки.

2 МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ, РЕШЕНИЯ ЗАДАЧ КОМПЬЮТЕРНОГО ЗРЕНИЯ

2.1 Анализ методов машинного обучения

Рассмотрим метод ансамблей. Случайный лес (Random Forest) – один из самых популярных методов прогнозной аналитики. Идея данного метода в том, что, если нам сложно построить прогнозную модель, которая обеспечивала бы желаемую точность на разнообразных входных данных, мы можем попробовать построить сразу несколько независимых моделей для нашей задачи, чтобы каждая из них обладала определенной точностью (пусть недостаточно высокой), но за счет определенной «рандомизации» эти модели ошибались бы на разных множествах данных. Тогда мы можем комбинировать результаты данных моделей, чтобы в среднем качество результата повысилось. То есть модели будут «сглаживать» ошибки друг друга. За общий результат такого ансамбля можно взять среднее значение или наиболее часто встречающееся. Строгого обоснования, почему это работает, не будет.

Алгоритм Random Forest является развитием идеи агрегации. Мы можем построить не одно дерево решений, а целый «лес» (forest). И если деревья достаточно различаются, в целом результаты работы модели должны быть лучше. Но помимо построения множества случайных выборок в Random Forest также в каждом узле дерева используется случайная выборка только части атрибутов из общего количества. Это также увеличивает случайность и сокращает время построения ансамбля. Количество деревьев в лесе можно увеличивать до тех пор, пока ошибка модели сокращается (100, 200, 500 деревьев и т.д.).

Алгоритм Random Forest реализован в Prognoz Platform. Он не доступен в интерфейсе пользовательских инструментов, но присутствует в библиотеке методов. Его можно использовать как для предсказания количественных атрибутов, так и для классификации данных по каким-либо категориям. Он прост

в использовании, обеспечивает высокую точность, имеет хорошую производительность, за счет чего и приобрел большую популярность.

Один из примеров применения Random Forest – оценка положения тела человека в пространстве контроллером Microsoft Kinect. Это контроллер для управления игровой консолью Xbox, позволяющий управлять персонажами игр при помощи движений своего тела, а также для голосового управления.

Далее с помощью обученного классификатора Random Forest для каждого пикселя получается вероятность принадлежности к определенной части тела. Чтобы такой классификатор работал точно для каждого пользователя, нужен очень большой набор данных для его «обучения». Сам процесс обучения длительный и занимает на таких наборах данных целые дни и недели процессорного времени. Но уже готовый алгоритм работает очень быстро. В данном случае – в реальном времени.

Рост производительности компьютеров и совершенствование алгоритмов позволяют решать задачи прогнозной аналитики все точнее и быстрее. Ансамбли моделей – это как раз одно из активно развивающихся направлений. Практика показала, что они относительно просты в обучении, дают существенный прирост точности, могут хорошо работать как с большими, так и с маленькими объемами данных. Этот метод, несомненно, займет свою нишу.

Рассмотрим метод Бэггинг (bootstrap aggregating) – это базовый метод построения ансамблей. Из исходного набора данных строится несколько равномерных случайных выборок такого же размера (например, 50, 100 и т.д.). Причем строки могут выбираться с повторами (а некоторые могут вообще не попасть). Это позволит обеспечить случайные вариации в результатах работы прогнозных моделей. На каждой выборке обучается прогнозная модель. При применении моделей их результаты агрегируются – берется наиболее часто встречающееся значение (голосование) или среднее значение.

Бустинг над решающими деревьями считается одним из наиболее эффективных методов с точки зрения качества классификации. Во многих экспериментах

наблюдалось практически неограниченное уменьшение частоты ошибок на независимой тестовой выборке по мере наращивания композиции. Более того, качество на тестовой выборке часто продолжало улучшаться даже после достижения безошибочного распознавания всей обучающей выборки. Это перевернуло существовавшие долгое время представления о том, что для повышения обобщающей способности необходимо ограничивать сложность алгоритмов. На примере бустинга стало понятно, что хорошим качеством могут обладать сколь угодно сложные композиции, если их правильно настраивать [12].

Впоследствии феномен бустинга получил теоретическое обоснование. Оказалось, что взвешенное голосование не увеличивает эффективную сложность алгоритма, а лишь сглаживает ответы базовых алгоритмов. Количественные оценки обобщающей способности бустинга формулируются в терминах отступа. Эффективность бустинга объясняется тем, что по мере добавления базовых алгоритмов увеличиваются отступы обучающих объектов. Причём бустинг продолжает раздвигать классы даже после достижения безошибочной классификации обучающей выборки.

Рассмотрим метод Стэкинг (Stacking). Стековое обобщение, или просто стекинг, – еще один способ объединения классификаторов, вводящий понятие мета-алгоритма обучения. В отличие от бэггинга и бустинга, при стекинге используются классификаторы разной природы. Идея стекинга такова:

– разбить обучающую выборку на два непересекающихся подмножества;

– обучить несколько базовых классификаторов на первом подмножестве;

– тестировать базовые классификаторы на втором подмножестве;

– используя предсказания из предыдущего пункта как входные данные, а истинные классы объектов как выход, обучить мета-алгоритм обучения.

Сверточная нейронная сеть (CNN) – это сетевая архитектура для глубокого обучения. CNN глубоко обученные искусственные нейронные сети [13], которые используются главным образом для классификации изображений, группируют их по сходству и выполняют распознавание объектов внутри сцен. CNN состоит из

одного или нескольких сверточных слоев и затем следует одним или несколькими полностью связанными слоями, как в стандартной многослойной нейронной сети.

Основным слоем является «Convolutional Layer» Сверточный слой. Основная задача которого- обнаружение локальных соединений объектов, из предыдущего слоя и сопоставление их внешнего вида с картой объектов. Далее следует «Non-Linearity Layer» Уровень нелинейности в сверточной нейронной сети состоит из функции активации, которая берет карту объектов, созданную сверточным слоем, и создает карту активации как его вывод. Далее «Rectification Layer». Уровень выпрямления –уровень выпрямления в сверточной нейронной сети выполняет поэлементную операцию абсолютного значения на входном объеме. Далее следует «Rectified Linear Units (ReLU)» выпрямленные линейные единицы, представляющие собой специальную функцию, которая объединяет слои нелинейности и выпрямления в сверточных нейронных сетях. Выпрямленная линейная единица (то есть пороговое значение в нуле) является кусочно-линейной функцией. Выпрямленные линейные единицы имеют три существенных преимущества в сверточных нейронных сетях по сравнению с традиционными логистическими или гиперболическими функциями активации. Выпрямленные линейные единицы эффективно распространяют градиент и, следовательно, уменьшают вероятность проблемы градиента, которая характерна для глубоких нейронных архитектур. Выпрямленные линейные единицы обнуляют отрицательные значения до нуля и, следовательно, решают проблему отмены, а также приводят к гораздо более разреженному объему активации на его выходе. Разреженность полезна по нескольким причинам, но в основном обеспечивает устойчивость к небольшим изменениям входных данных, таким как шум. Выпрямленные линейные единицы состоят только из простых операций с точки зрения вычислений (в основном сравнения) и поэтому намного более эффективны для реализации в сверточных нейронных сетях. Благодаря своим преимуществам и производительности, большинство современных архитектур сверточных нейронных сетей используют только выпрямленные линейные единичные слои

(или их производные, такие как ReLU с шумом или утечкой) в качестве своих уровней нелинейности вместо традиционных уровней нелинейности и выпрямления. «Pooling Layer» Слой пула: слой пула или нисходящей выборки отвечает за уменьшение специального размера карты активации. «Fully Connected Layer» полностью связанный слой. Полностью связанные уровни в сверточной сети практически многослойный персептрон (как правило, двух или трехслойный MLP), который направлен на отображение объема активации из комбинации предыдущих различных слоев в классовое распределение вероятностей. «Dropout Layer» слой Dropout – это метод, используемый для защиты от переобучения в нейронных сетях, главная идея Dropout – вместо обучения одной DNN обучить ансамбль нескольких DNN, а затем усреднить полученные результаты. Softmax: функция Softmax рассчитывает распределение вероятностей события по «n» различным событиям. В общем, эта функция будет вычислять вероятности каждого целевого класса по всем возможные целевые классы. Позже рассчитанные вероятности будут полезны для определения нового целевого класса для заданных входов.

В дальнейшем, чтобы найти самую низкую ошибку (глубочайшую впадину) в функции потерь (по отношению к одному весу), нужно настроить параметры модели. В этом поможет математический анализ. Благодаря анализу мы знаем, что наклон графика функции – производная от функции по переменной. Это наклон всегда указывает на ближайшую впадину.

Функция потерь предназначена для отслеживания ошибки с каждым примером обучениям, в то время как производная функции относительного одного веса – это то, куда нужно сместить вес, чтобы минимизировать ее для этого примера обучения. Вы можете создавать модели даже без применения функции потерь. Но вам придется использовать производную относительно каждого веса. Теперь, когда мы определили направление, в котором нужно двигаться, нам нужно понять, как это сделать. Тут мы используем коэффициент скорости обучения, его называют гипер-параметром. Гипер-параметр – значение, требуемое вашей моделью, о котором мы действительно имеем очень смутное представление. Обычно эти

значения могут быть изучены методом проб и ошибок. Здесь не так: одно подходит для всех гипер-параметров. Коэффициент скорости обучения можно рассматривать как «шаг в правильном направлении. Это была функция потерь, построенная на один вес. В реальной модели мы делаем всё перечисленное выше для всех весов, перебирая все примеры обучения. Даже в относительно небольшой модели машинного обучения у вас будет более чем 1 или 2 веса. Это затрудняет визуализацию, поскольку график будет обладать очень большими размерами.

Для дальнейшего исследования будем исползать метод нейронных сетей так как он является наиболее подходящим для нашего исследования.

2.2 Нейронные сети

Сверточные нейронные сети – это глубокие искусственные нейронные сети, которые используются, главным образом, для классификации изображений, кластеризуют их по сходству (поиск фотографий) и выполняют распознавание объектов внутри сцен. С помощью них можно реализовать алгоритмы, которые могут идентифицировать лица, выделить отдельные лиц и распознать их, различать уличные знаки, опухоли, животных, растения, автотранспорт, авто номера и многие другие элементы визуальных данных.

Когда мы говорим о сверточной нейронной сети (CNN), мы обычно подразумеваем о компьютерном видении. CNN были ответственны за крупные прорывы в классификации изображений и сегодня являются ядром большинства систем компьютерного зрения, от автоматической маркировки фотографий Facebook до автомобилей с автоматическим управлением.

Сверточные сети выполняют распознавание символов (OCR) для оцифровки текста и делают возможной обработку рукописных документов, где изображения являются обычной фотографией. СНС также могут быть применены к звуку, когда он визуально представлен в виде спектрограммы. Совсем недавно сверточные сети применялись непосредственно к рукописному анализу текста, а также к графикам и диаграммам.

Эффективность сверточных сетей в распознавании изображений является одной из основных причин эффективности машинного обучения. Они поддерживают основные достижения в области компьютерного зрения (CV), который имеет очевидные применения для автомобилей с беспилотным управлением, робототехники, систем летательных аппаратов, систем безопасности, медицинской диагностики и лечения слабовидящих.

Сверточные нейронные сети принимают и обрабатывают изображения как тензоры, а тензоры представляют собой матрицы чисел с дополнительными измерениями.

Их сложно представить, поэтому давайте рассмотрим их по аналогии. Скаляр – это просто число, например 7; вектор представляет собой список чисел (например, 7,8,9); и матрица - это прямоугольная сетка чисел, занимающая несколько строк и столбцов, как электронная таблица. Геометрически, если скаляр – это нульмерная точка, то вектор – это одномерная линия, матрица – это двумерная плоскость, стек матриц – это трехмерный куб, и когда каждый элемент этих матриц имеет стопка карт объектов привязана к нему, вы входите в четвертое измерение. Для примера матрица 2x2. Тензор охватывает размеры за этой двумерной плоскостью. Вы можете легко изобразить трехмерный тензор с массивом чисел, расположенных в кубе. На рисунке 1 тензор 2x3x2, представленный категорически (представьте нижний элемент каждого двухэлементного массива, проходящего вдоль оси z, чтобы интуитивно понять, почему он называется 3-мерным массивом):

$$\begin{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 5 \end{pmatrix} & \begin{pmatrix} 4 \\ 7 \end{pmatrix} \\ \begin{pmatrix} 3 \\ 4 \end{pmatrix} & \begin{pmatrix} 4 \\ 6 \end{pmatrix} & \begin{pmatrix} 5 \\ 8 \end{pmatrix} \end{pmatrix}$$

Рисунок 1 – Тензор 2x3x2

В коде приведенный выше тензор (рисунок 1) будет выглядеть следующим образом: [[[2,3],[3,5],[4,7]],[[3,4],[4,6],[5,8]]].

Визуально он представлен на рисунке 2:

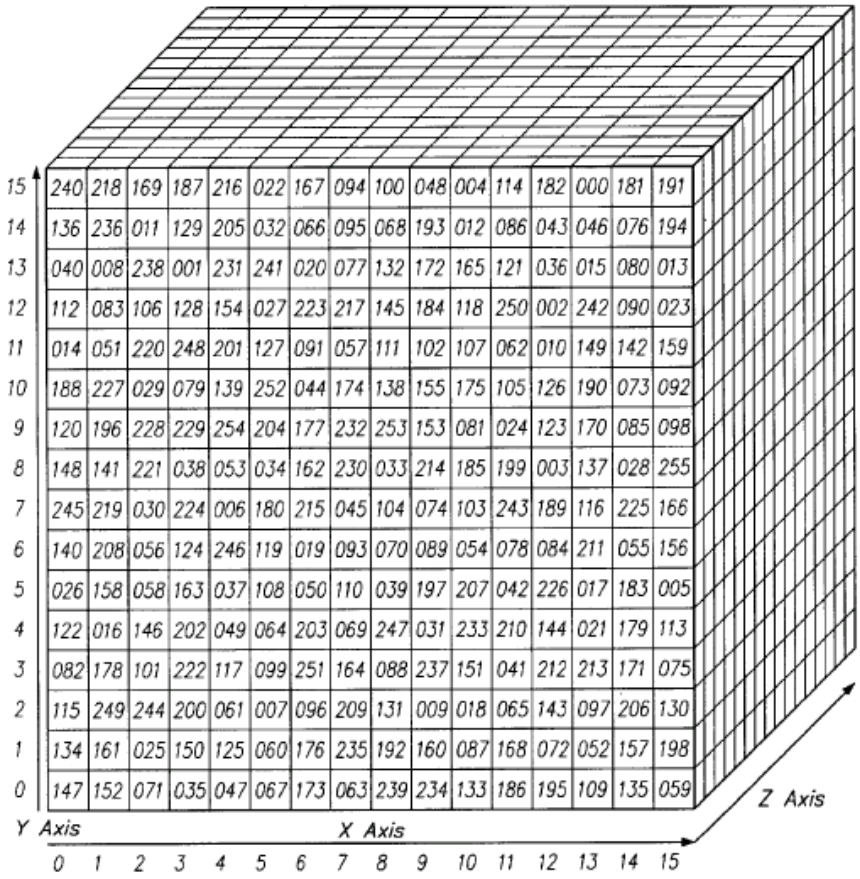


Рисунок 2 – Матрица

Другими словами, тензоры образованы массивами, вложенными в массивы, и это вложение может продолжаться бесконечно, составляя произвольное число измерений, намного превышающее то, что мы можем визуализировать в пространстве. 4-D тензор просто заменит каждый из этих скаляров на массив, вложенный на один уровень глубже. Сверточные сети имеют дело с 4-мерными тензорами, как показано ниже на рисунке 3.

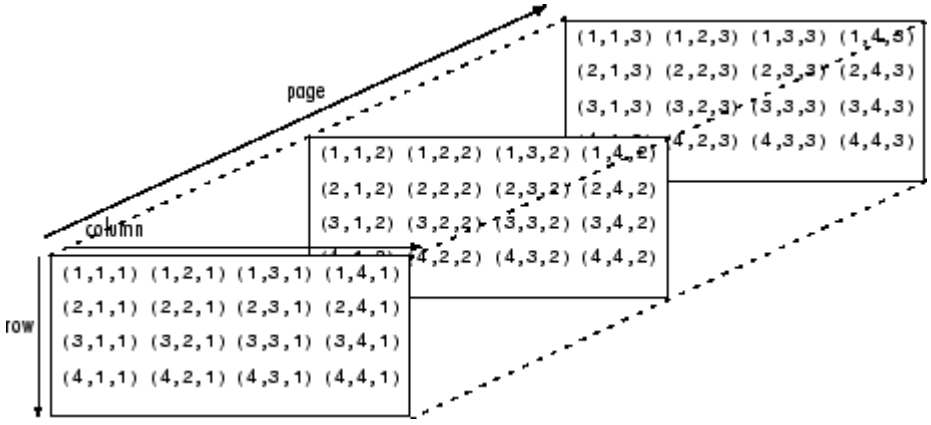


Рисунок 3 – Тензорный слой

Размерность тензора (1,2,3…n) называется его порядком; тензор пятого порядка будет иметь пять измерений. Глубина необходима для того, чтобы закодировать цвета. Например, кодирование Red-Green-Blue (RGB) создает изображение глубиной в три слоя. Каждый слой называется «каналом», и благодаря свертке он создает стопку карт объектов, которые существуют в четвертом измерении, прямо по улице от самого времени (функции – это просто детали изображений, например, линия или кривая, для которых сверточные сети создают карты).

Таким образом, вместо того, чтобы рассматривать изображения как двумерные области, в сверточных сетях они рассматриваются как четырехмерные объемы.

Для математических целей свертка – это интеграл, измеряющий насколько одна функция перекрывает другую функцию, когда одна проходит над другой (рисунок 4).

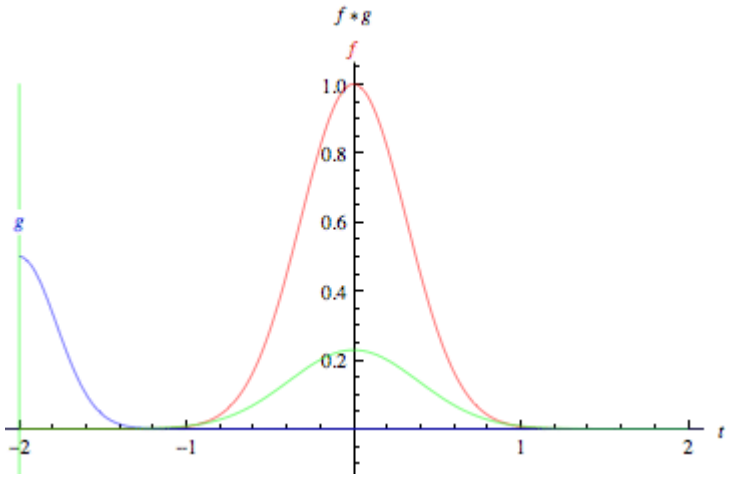


Рисунок 4 – Прохождение функций

Интеграл – это площадь высокой кривой «колокола», стоящую посередине графика. Рядом с ним находится вторая кривая колокола, которая короче и шире, медленно дрейфует с левой стороны графика вправо. Продуктом перекрытия этих двух функций в каждой точке вдоль оси x является их свертка. Таким образом, в некотором смысле, эти две функции «объединяются».

При анализе изображений статическая базовая функция (эквивалент кривой неподвижного «колокола») является анализируемым входным изображением, а

вторая мобильная функция называется фильтром, поскольку она улавливает сигнал или функцию в изображении. Эти две функции связаны посредством умножения.

Сверточные сети пропускают множество фильтров по одному изображению, каждый из которых получает свой сигнал. На начальном слое представим их как фильтр в виде горизонтальной линии, вертикальной линии и диагональной линии, чтобы создать карту краев на изображении.

Сверточные сети берут эти фильтры, часть пространственного объекта изображения и отображают их один за другим; то есть они создают карту каждого места, где встречается объект. Сверточные сети выполняют своего рода поиск. Представим маленькое увеличительное стекло, скользящее слева направо по большому изображению, оказавшись в конце изображения, оно возобновляется с новой строки (по принципу работы печатной машинки). Это движущееся окно способно распознавать только один признак, допустим, короткую вертикальную линию. Три темных пикселя сложены друг на друга. Она (линза) перемещает этот распознающий вертикальный фильтр по фактическим пикселям изображения в поисках совпадений.

Каждый раз, когда найдено совпадение, оно сопоставляется с пространством объектов, специфичным для этого визуального элемента. В этом месте записывается местоположение каждого совпадения вертикальной линии. Сверточная сеть выполняет столько поисков по горизонтальным, диагональным и вертикальным линиям по одному изображению, сколько необходимо найти визуальных элементов.

После сверточного слоя входные данные передаются через нелинейное преобразование, такое как функция активации tanh или выпрямленная линейная единица, которая будет свертывать входные значения в диапазон от -1 до 1.

2.3 Обучение сверточной нейронной сети

Сверточные сети воспринимают изображения иначе чем люди. Необходимо учитывать этот факт при выборе изображения для обработки сверточной сетью.

Сверточные сети воспринимают изображения как объемы; т.е. трехмерные объекты, а не плоские полотна, измеряемые только по ширине и высоте. Это связано с тем, что цифровые цветные изображения имеют красно-сине-зеленую (RGB) кодировку, смешивая эти три цвета для получения цветового спектра, который воспринимает человеческий мозг. Сверточная сеть принимает такие изображения в виде трех отдельных слоев цвета, расположенных один над другим.

Таким образом, сверточная сеть получает нормальное цветное изображение в виде прямоугольника, ширина и высота которого измеряются количеством пикселей вдоль этих измерений, а глубина составляет три слоя, по одному на каждую букву в RGB. Эти глубинные слои называются каналами. По мере того как изображения перемещаются по сверточной сети, математически выражая их в виде матриц нескольких измерений (30x30x3). Необходимо уделять пристальное внимание точным показаниям каждого измерения объема изображения, поскольку они являются основой операций линейной алгебры, которые используются для обработки изображений. Теперь для каждого пикселя изображения интенсивность R, G и B будет выражаться числом, и это число будет элементом в одной из трех сложенных двумерных матриц, которые вместе образуют объем изображения. Эти числа являются исходными, необработанными сенсорными признаками, поступающими в сверточную сеть. Цель ConvNets состоит в том, чтобы найти, какие из этих чисел являются значимыми сигналами, которые фактически помогают ему более точно классифицировать изображения. Вместо того, чтобы фокусироваться на одном пикселе за раз, сверточная сеть берет квадратные участки пикселей и пропускает их через фильтр. Этот фильтр также представляет собой квадратную матрицу, меньшую, чем само изображение, и равное по размеру патчу, также называется ядром.

Рассмотрим две матрицы:

1) матрица размером 30x30;
2) матрица размером 3x3.

То есть фильтр второй матрицы, покрывает одну сотую площади поверхности одного канала изображения. Рассмотрим точечное произведение фильтра с этим патчем канала изображения. Если две матрицы имеют высокие значения в одинаковых позициях, выход точечного произведения будет высоким, иначе значение будет низким. Таким образом, одно значение – выход точечного произведения – показывает, соответствует ли рисунок пикселя в базовом изображении образцу пикселя, выраженному данным фильтром. Давайте представим, что фильтр показывает горизонтальную линию с высокими значениями во второй строке и низкими значениями в первой и третьей строках. Фильтр начинает движение в верхнем левом углу базового изображения, и шаг за шагом перемещается по изображению, пока он не достигнет верхнего правого угла. Размер шага известен как шаг. Фильтр возможно переместить вправо на один столбец за раз, или установить более крупные шаги. На каждом шаге рассматриваем точечный продукт и фиксируем результаты этого точечного продукта в третью матрицу, известную как карта активации. Ширина или количество столбцов карты активации равно числу шагов, которые фильтр выполняет для прохождения базового изображения. Так как большие шаги приводят к меньшему количеству шагов, большой шаг даст меньшую карту активации. Это важно, потому что размер матриц, которые сверточные сети обрабатывают и производят на каждом уровне, прямо пропорционален их вычислительным затратам и времени, затрачиваемому на обучение. Больший шаг означает меньше времени и вычислений. Фильтр, наложенный на первые три строки, будет скользить по ним, а затем снова начнется со строк 4–6 того же изображения. Если он имеет шаг три, то он произведет матрицу точечных произведений, которая будет равна матрице 10x10. Тот же фильтр, представляющий горизонтальную линию, может быть применен ко всем трем каналам базового изображения, R, G и B. И три карты активации 10x10 могут быть добавлены вместе, так что совокупная карта активации для горизонтальной линии по всем трем каналам исходного изображения также 10x10. Поскольку у

изображений есть линии, идущие во многих направлениях, и они содержат много разных видов фигур и узоров пикселей, нужно пройти другими фильтрами по базовому изображению в поисках этих рисунков. Например, рассмотрим поиск 96 различных шаблонов в пикселях. Эти 96 шаблонов создадут стопку из 96 карт активации, в результате чего получится новый том 10x10x96. На рисунке 5 произведена замена входного изображения, ядра и карты активации вывода.

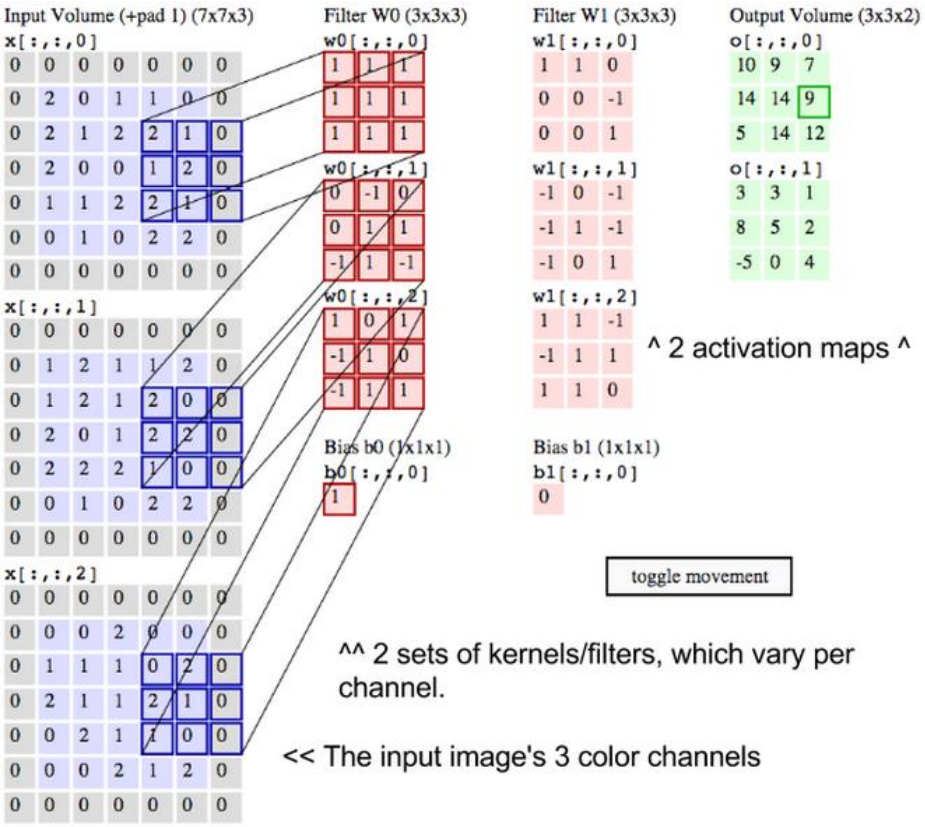


Рисунок 5 – Ядра и карты активации вывода

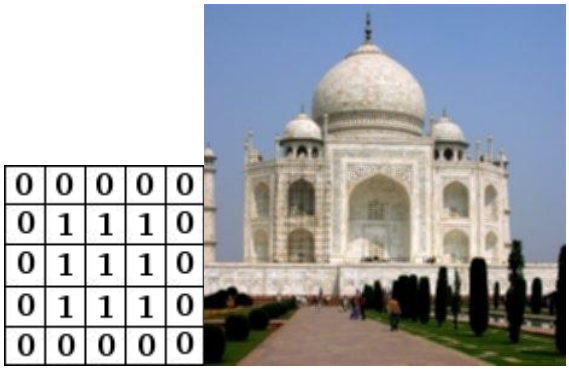
Этот процесс называется сверткой. Свертку можно представить как замысловатый вид умножения, используемый при обработке сигналов. Так же второй способ представить две матрицы, создающие скалярное произведение, как две функции. Изображение – это основная функция, а фильтр – это функция, которую можно представить, как наложение на основную функцию (рисунок 7).

Одна из основных проблем с изображениями состоит в том, что они являются многомерными, следовательно, они требуют много времени и вычислительной мощности для обработки. Сверточные сети предназначены для уменьшения

размерности изображений различными способами. Шаг фильтра – это один из способов уменьшить размерность.

Предположим, что матрица слева представляет черно-белое изображение. Каждая запись соответствует одному пикселю, 0 для черного и 1 для белого (обычно это от 0 до 255 для изображений в оттенках серого). Скользящее окно называется детектором ядра, фильтра или объекта. Рассмотрим фильтр 3×3, умножаем его значения поэлементно на исходную матрицу, а затем суммируем их. Чтобы получить полную свертку, необходимо проделать это для каждого элемента, перемещая фильтр по всей матрице.

Рассмотрим на практике. Усреднение каждого пикселя с соседними значениями размывает изображение (рисунок 6).



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Рисунок 6 – исходное изображение

В частности, взяв разницу между пикселем и соседним пикселем, обнаруживаются ребра, это происходит в гладких частях изображения, где цвет пикселя равен цвету его соседей: добавления отменяются, и результирующее значение равно 0, или черному. Если есть резкий край по интенсивности (например, при переходе от белого к черному), мы получаем большую разницу и значение белого.

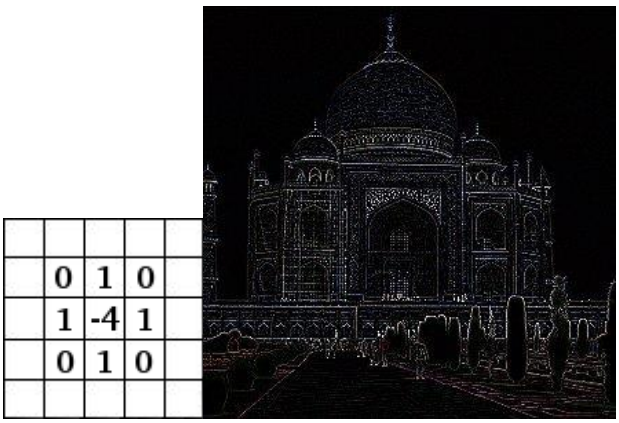


Рисунок 7 – изображение с усредненными значениями соседних пикселей

Принцип CNN – это всего лишь несколько слоев сверток с нелинейными функциями активации, такими как ReLU или tanh, которые применяются к результатам. В традиционной нейронной сети с прямой связью подключается каждый входной нейрон к каждому выходному нейрону в следующем слое. Это также называется полностью связанным слоем. В данном случае, CNN этого не подразумевает. Вместо этого используются свертки над входным слоем для вычисления выходных данных. Это приводит к локальным соединениям, где каждая область входа связана с нейроном на выходе. Каждый слой применяет различные фильтры, обычно сотни или тысячи, и объединяет их результаты. На этапе обучения CNN автоматически запоминает значения своих фильтров в зависимости от задачи, которую мы хотим выполнить. Например, в классификации изображений CNN может научиться обнаруживать края из необработанных пикселей в первом слое, затем использовать края для обнаружения простых форм во втором слое, а затем использовать эти формы для определения объектов более высокого уровня, таких как формы лица. в более высоких слоях. Последний слой является классификатором, который использует эти высокоуровневые функции.

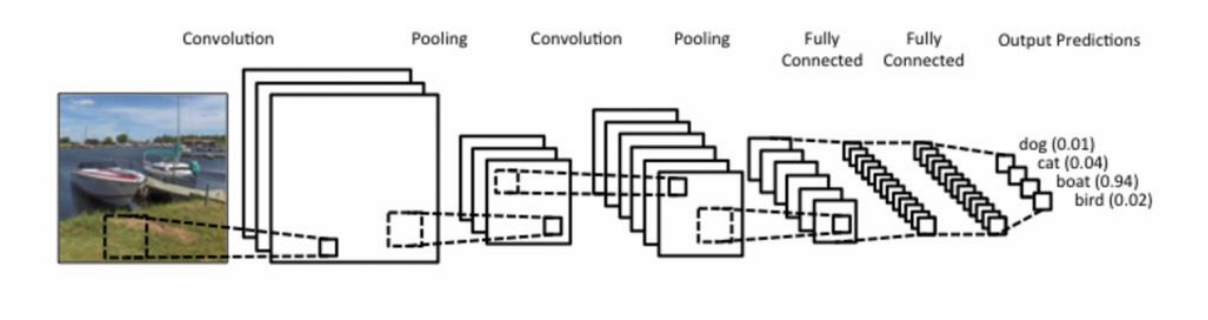


Рисунок 8 – Пример слоев сверточной нейронной сети.

Есть два аспекта этого вычисления, на которые стоит обратить внимание: инвариантность местоположения и композиционность. Допустим необходимо классифицировать, есть ли на изображении слон. Поскольку фильтр скользит по всему изображению, ему не важно, где находится слон. На практике фильтру также дает неизменность для перемещения, вращения и масштабирования. Второй ключевой аспект – (локальная) композиционность. Каждый фильтр объединяет локальный патч низкоуровневых объектов в высокоуровневое представление. Вот почему CNN такие мощные в компьютерном зрении. Понятно, что мы строим ребра из пикселей, фигуры из ребер и более сложные объекты из фигур. Вместо пикселей изображения входными данными для большинства задач являются предложения или документы, представленные в виде матрицы. Каждая строка матрицы соответствует одному токену, обычно слову, но это может быть символ. То есть каждая строка – это вектор, представляющий слово. Как правило, эти векторы представляют собой вложения слов (низкоразмерные представления), но они также могут быть горячими векторами, которые индексируют слово в словарь. Для предложения из 10 слов, использующего 100-мерное вложение, представим матрицу 10×100 в качестве входных данных. Фильтры скользят по локальным участкам изображения, но обычно используются фильтры, которые скользят по полным строкам матрицы (словам). Таким образом, «ширина» фильтров обычно равна ширине входной матрицы. Высота или размер могут варьироваться, но типичным является сдвиг окна на 2–5 слов за раз. Собрав все вышеперечисленное, CHC может выглядеть следующим образом (рисунок 8).

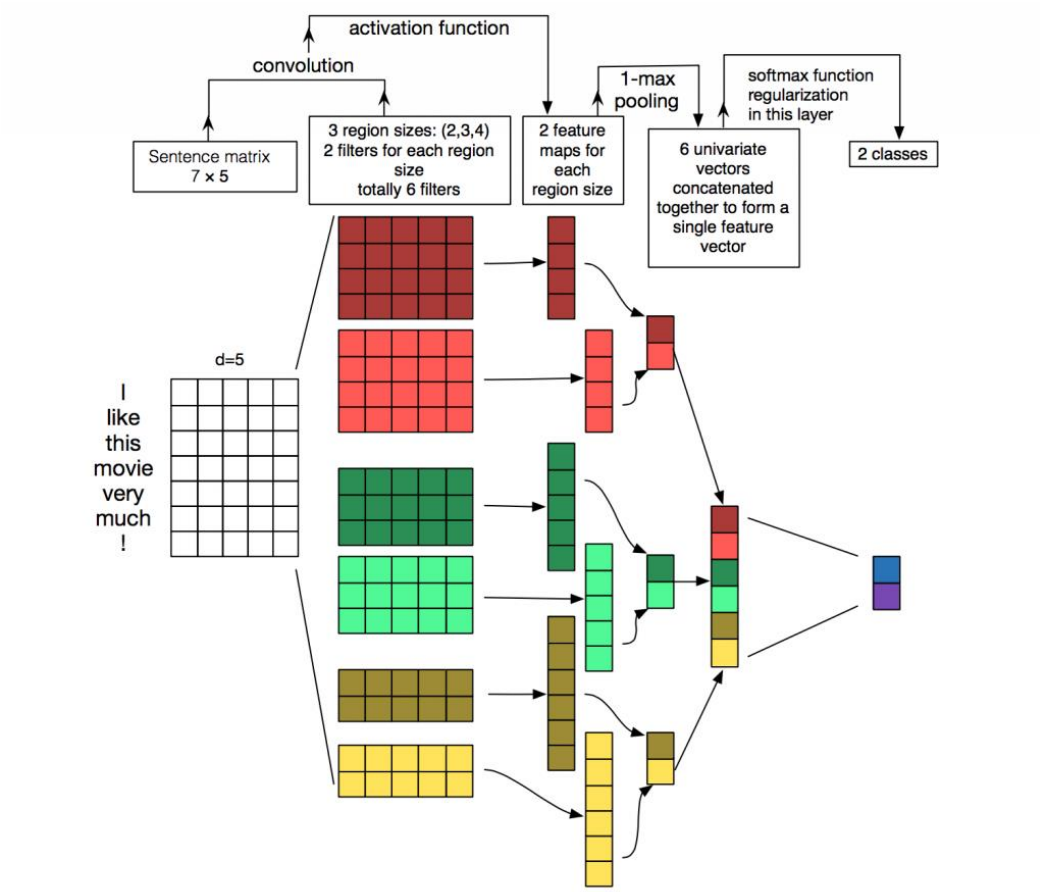


Рисунок 9 – Иллюстрация архитектуры сверточной нейронной сети.

Иллюстрация архитектуры сверточной нейронной сети, CNN (рисунок 9), для классификации предложений. Изобразим три размера области фильтра: 2, 3 и 4, каждый из которых имеет 2 фильтра. Каждый фильтр выполняет свертку на матрице предложений и генерирует карты характеристик (переменной длины). Затем выполняется пул 1-макс для каждой карты, т.е. записывается наибольшее число из каждой карты объектов. Таким образом, из всех шести карт генерируется одномерный вектор объектов, и эти 6 объектов объединяются, чтобы сформировать вектор объектов для предпоследнего слоя. Окончательный слой softmax затем получает этот вектор объектов в качестве входных данных и использует его для классификации предложения; здесь предполагаем двоичную классификацию и, следовательно, изображаем два возможных состояния выхода. Инвариантность местоположения и локальная композиционность имеют интуитивный смысл для изображений. Очень важно, как в предложении появляется слово. Пиксели, близкие

друг к другу, могут быть семантически связаны (часть одного и того же объекта), но это не всегда верно для слов. Во многих языках части фраз могут быть разделены несколькими другими словами. Композиционный аспект также не очевиден. Очевидно, что слова сочетаются в некотором роде, как прилагательное, изменяющее существительное, но как именно это работает, что на самом деле означают представления более высокого уровня, не так очевидно, как в случае с компьютерным зрением. Рекуррентные нейронные сети имеют более интуитивный смысл. Они напоминают то, как происходит обработка языка (или, по крайней мере, то, как человек думает, что обрабатывает язык) чтение последовательно слева направо. Это не значит, что CNN не работают. Простая модель Bag of Words – явное упрощение с неверными допущениями, но, тем не менее, в течение многих лет являлась стандартным подходом и приводила к довольно хорошим результатам. Весомым аргументом является для CNN то, что они быстрые. Свертки возникают центральной частью компьютерной графики и реализуются на аппаратном уровне на графических процессорах. По сравнению с чем-то вроде n-граммов , CNN также эффективны с точки зрения представления. Сверточные фильтры выучивают хорошие представления автоматически, без необходимости представлять весь словарный запас. Вполне разумно иметь фильтры размером больше 5. Многие из изученных фильтров в первом слое занимают функции, весьма похожие, но не ограничивающиеся на n-граммы, и представляют их более компактным способом. Прежде чем объяснить, как CNN применяются к задачам обработки естественного языка, давайте рассмотрим некоторые варианты, которые необходимо сделать при создании CNN. Применение фильтра 3×3 в центре матрицы работает нормально, однако обратим внимание на края. Можем использовать заполнение нулями. Все элементы, которые выходят за пределы матрицы, принимаются равными нулю. Делая это, мы можем применить фильтр к каждому элементу нашей входной матрицы и получить больший или одинаковый размер вывода. Добавление дополнения нулями также называется широкой сверткой, и отсутствие использования дополнения нулями было бы узкой сверткой.

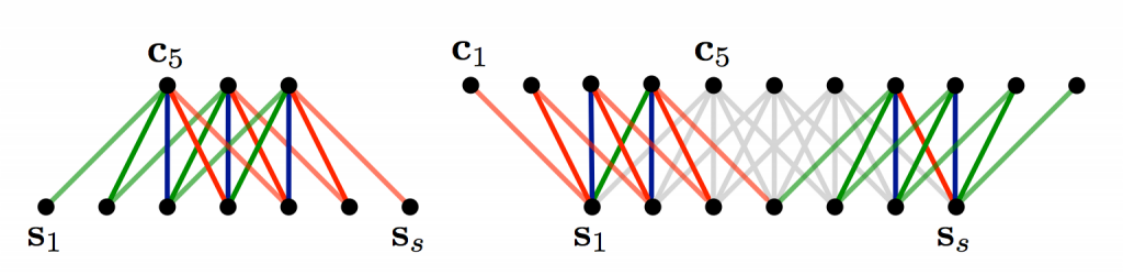


Рисунок 10 – Узкая и широкая свертка. Размер фильтра 5, входной размер 7.

Вы можете увидеть (рисунок 10,11), насколько широкая свертка полезна или даже необходима, когда большой фильтр относительно размера ввода. Еще один параметр для сверток – это размер шага, определяемый тем, насколько вы хотите смещать фильтр на каждом шаге. Во всех приведенных выше примерах размер шага составлял 1, и последовательные применения фильтра перекрывались. Больший размер шага приводит к меньшему количеству применений фильтра и меньшему выходному размеру.

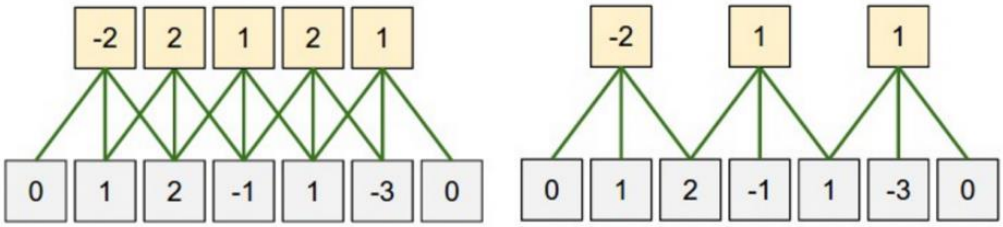


Рисунок 11 – Размер сверточного шага. Слева: размер шага 1. Справа: размер шага 2

В литературе мы обычно видим размеры шага 1, но больший размер шага может позволить вам построить модель, которая ведет себя примерно так же, как рекурсивная нейронная сеть, то есть выглядит как дерево.

Ключевым аспектом сверточных нейронных сетей являются уровни объединения, обычно применяемые после сверточных слоев. Наиболее распространенный способ сделать это, чтобы применить максимальное обьединение на результат каждого фильтра. Не обязательно нужно объединяться по всей матрице, вы также можете объединиться лишь часть. Например, ниже

показан максимальный пул для окна 2×2 (в NLP мы обычно применяем пул по всему выводу, получая только одно число для каждого фильтра), рисунок 12.

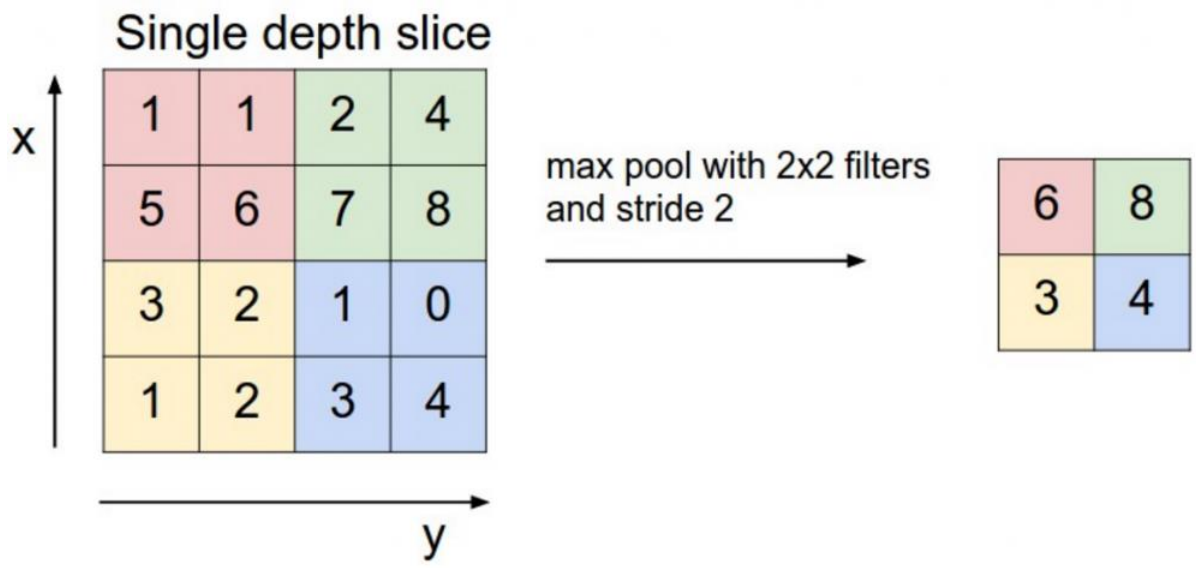


Рисунок 12 – Максимизированное агрегирование свертки 2x2

Есть несколько причин объединения. Одним из свойств пула является то, что он обеспечивает выходную матрицу фиксированного размера, которая обычно требуется для классификации. Например, если у вас есть 1000 фильтров, и вы применяете максимальный пул к каждому, вы получите 1000-мерный вывод, независимо от размера ваших фильтров или размера вашего ввода. Это позволяет вам использовать входные данные переменного размера и фильтры переменного размера, но всегда получать одинаковые выходные размеры для дальнейшей классификации. Объединение в пул также уменьшает размерность выходных данных, но сохраняет наиболее важную информацию. Для простоты понимания о каждом фильтре можно думать, как о любом слове, и при обнаружении, если это слово содержится где-то в предложении, результат применения фильтра к этой области даст большое значение, и низкое значение в других зонах. Выполняя операцию максимального объединения, вы сохраняете информацию о том, появилась ли наше слово в предложении, но вы теряете информацию о том, где именно оно появилось. Когда вы объединяете область, выходной сигнал останется примерно таким же, даже если вы сместите или поверните изображение на

несколько пикселей, поскольку операция максимального объединения выбирают одно и то же значение независимо.

Каналы – это разные «просмотры» ваших входных данных. Например, при распознавании изображений у вас обычно есть RGB (красный, зеленый, синий) каналы. Вы можете применять свертки по каналам, с разными или равными весами. Также можете представить себе разные каналы: у вас могут быть отдельные каналы для разных вложений слов, или у вас может быть канал для одного и того же предложения, представленного на разных языках.

Наиболее естественным подходом для CNN являются задачи классификации, такие как анализ эмоций, обнаружение спама или категории тем. Из-за свертки и операции объединения в пул теряют информацию о локальном порядке слов, поэтому для выхода из данной ситуации сохраняют последовательность с помощью, как в PoS Tagging или Entity Extraction[14].

Рассмотрим переобучение нейронной сети. Переобучение о это явление, при котором ИНС изучает сложные входные выходные отношения, которые специфичны только для данных обучения и, таким образом, перестают фиксировать истинное отношение между входами и выходами. Другими словами, ИНС начинает накладывать дополнительные ограничения и ограничения, полученные из подмножества данных, которые не всегда удовлетворяются условиям обучения. Примером может являться сеть, которая ожидает, что автомобиль будет иметь четыре двери и крышу, поскольку она была обучена в основном на изображениях автомобилей с четырьмя дверями и крышей. Здесь обсуждаются три наиболее популярных метода. Первый способ ограничить веса от более высоких значений, что предотвращает любое конкретное подмножество входных данных для имеющих больший вес при определении выходов с помощью метода, называемого регуляризацией. Второй метод использует несколько сетевых архитектур для обучения по тем же данным и выбирая окончательный результат через схему усреднения или голосования. В третьем методе искусственно генерируется больше данных обучения с использованием методов расширения

обучающего подмножества. Идея состоит в том, чтобы показать больше изменений входов в сеть. Каждый из этих методов кратко обсуждается ниже[16] .

2.4 Функции активации нейросети

Функция активации определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения. Теперь значение *Y* может быть любым в диапазоне от минус бесконечности до плюс бесконечности. В действительности нейрон не знает границу, после которой следует активация. Ответим на вопрос, как мы решаем, должен ли нейрон быть активирован (мы рассматриваем паттерн активации, так как можем провести аналогию с биологией. Именно таким образом работает мозг, а мозг – хорошее свидетельство работы сложной и разумной системе).

Для этой цели решили добавлять активационную функцию. Она проверяет произведенное нейроном значение *Y* на предмет того, должны ли внешние связи рассматривать этот нейрон как активированный, или его можно игнорировать.[16]

Ступенчатая функция активации

Рассмотрим вопрос о том, что считать границей активации для активационной функции. Если значение *Y* больше некоторого порогового значения, считаем нейрон активированным. В противном случае считаем, что нейрон неактивен. Такая схема должна сработать, но сначала давайте её формализуем:

- функция A = активирована, если Y > граница, иначе нет;
- другой способ: A = 1, если Y > граница, иначе A = 0.

Функция, которую мы только что создали, называется ступенчатой. Такая функция представлена на рисунке 13.

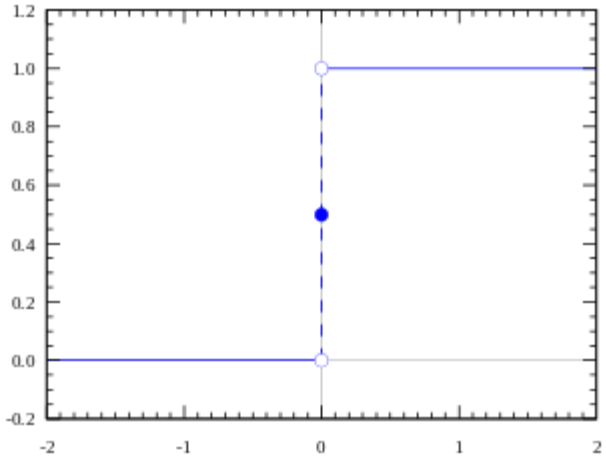


Рисунок 13 – Иллюстрация функции L

Функция принимает значение *1* (активирована), когда *Y>0* (граница), и значение 0 (не активирована) в противоположном случае. Рассмотрим активационную функцию для нейрона. Это простая функция, однако в неё есть недостатки. Представим, что создаётся бинарный классификатор – модель, которая должна говорить «да» или «нет» (активирован или нет). Ступенчатая функция в точности выводит 1 или 0. Но если появляется среднее значение, то данная функция округлит его в большую или меньшую сторону.

Теперь представим случай, когда требуется большее количество нейронов для классификации многих классов: класс 1, класс 2, класс 3 и так далее. Что будет, если активированными окажутся больше, чем 1 нейрон. Все нейроны из функции активации выведут 1. В таком случае появляются вопросы о том, какой класс должен в итоге получиться для заданного объекта.

Мы хотим, чтобы активировался только один нейрон, а функции активации других нейронов были равна нулю (только в этом случае можно быть уверенным, что сеть правильно определяет класс). Такую сеть труднее обучать и добиваться сходимости. Если активационная функция не бинарная, то возможны значения «активирован на 50 %», «активирован на 20 %» и так далее. Если активированы несколько нейронов, можно найти нейрон с наибольшим значением активационной функции (лучше, конечно, чтобы это была softmax функция, а не max. Но в таком случае, как и ранее, если более одного нейрона говорят «активирован на 100 %»,

проблема по-прежнему остается. Так как существуют промежуточные значения на выходе нейрона, процесс обучения проходит более гладко и быстро, а вероятность появления нескольких полностью активированных нейронов во время тренировки снижается по сравнению со ступенчатой функцией активации (хотя это зависит от того, что вы обучаете и на каких данных).

На рисунке 14 представлена линейная функция активации.

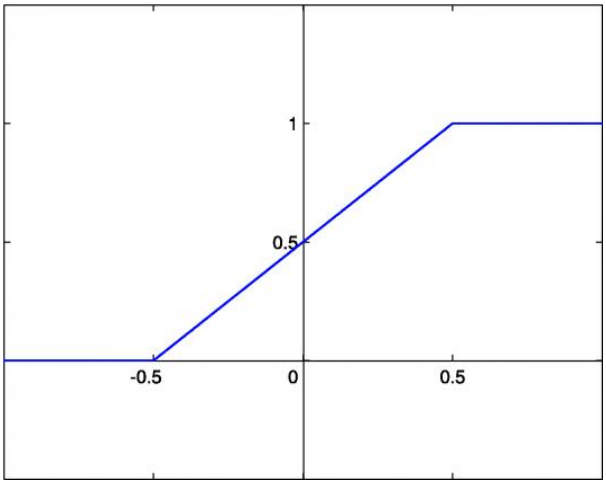


Рисунок 14 – Линейная функция активации

Линейная функция представляет собой прямую линию и пропорциональна входу (то есть взвешенной сумме на этом нейроне).

Такой выбор активационной функции позволяет получать спектр значений, а не только бинарный ответ. Можно соединить несколько нейронов вместе и, если более одного нейрона активировано, решение принимается на основе применения операции max (или softmax).

Производная от $A=cx$ по x равна c. Это означает, что градиент никак не связан с X. Градиент является постоянным вектором, а спуск производится по постоянному градиенту. Если производится ошибочное предсказание, то изменения, сделанные с обратным распространением ошибки, тоже постоянны и не зависят от изменения на входе. Но существует и другая проблема. Каждый слой активируется линейной функцией. Значение с этой функции идет в следующий слой в качестве входа, второй слой считает взвешенную сумму на своих входах и, в свою очередь, включает нейроны в зависимости от другой линейной

активационной функции. Не имеет значения, сколько слоев мы имеем. Если все они по своей природе линейные, то финальная функция активации в последнем слое будет просто линейной функцией от входов на первом слое. Это означает, что два слоя (или N слоев) могут быть заменены одним слоем. Мы потеряли возможность делать наборы из слоев. Не важно, как мы стэкаем, вся нейронная сеть все равно будет подобна одному слою с линейной функцией активации (комбинация линейных функций линейным образом – другая линейная функция).

Рассмотрим функцию активации Сигмоида, которая представлена на рисунке 15.

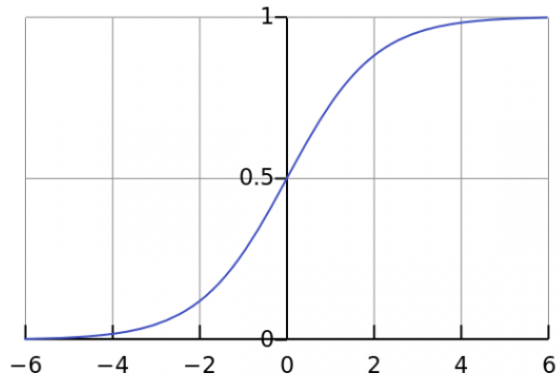


Рисунок 15 – Функция Сигмоиды

Сигмоида – нелинейна по своей природе, а комбинация таких функций производит также нелинейную функцию. Теперь мы можем стэкать слои.

Еще одно достоинство такой функции – она не бинарна, что делает активацию аналоговой, в отличие от ступенчатой функции. Для сигмоиды также характерен гладкий градиент.

Если вы заметили, в диапазоне значений *X* от -2 до 2 значения Y меняется очень быстро. Это означает, что любое малое изменение значения *X* в этой области влечет существенное изменение значения *Y*. Такое поведение функции указывает на то, что Y имеет тенденцию прижиматься к одному из краев кривой.

Сигмоида действительно выглядит подходящей функцией для задач классификации. Она стремиться привести значения к одной из сторон кривой (например, к верхнему при $x=2$и нижнему при $x=-2$). Такое поведение позволяет находить четкие границы при предсказании.

Другое преимущество сигмоиды над линейной функцией заключается в следующем. В первом случае имеем фиксированный диапазон значений функции – [0,1], тогда как линейная функция изменяется в пределах (-inf, inf). Такое свойство сигмоиды очень полезно, так как не приводит к ошибкам в случае больших значений активации.

Сегодня сигмоида является одной из самых частых активационных функций в нейросетях. Но и у неё есть свои недостатки. Вы уже могли заметить, что при приближении к концам сигмоиды значения Y имеют тенденцию слабо реагировать на изменения в X. Это означает, что градиент в таких областях принимает маленькие значения. А это, в свою очередь, приводит к проблемам с градиентом исчезновения. Рассмотрим подробно, что происходит при приближении активационной функции к почти горизонтальной части кривой на обеих сторонах.

В таком случае значение градиента мало или исчезает (не может сделать существенного изменения из-за чрезвычайно малого значения). Нейросеть отказывается обучаться дальше или делает это крайне медленно (в зависимости от способа использования или до тех пор, пока градиент/вычисление не начнет страдать от ограничений на значение с плавающей точкой). Существуют варианты работы над этими проблемами, но даже с ними сигмоида остается всё ещё одна из самых популярных функций активации для задач классификации.

Рассмотри функцию активации – гиперболический тангенс, еще одна часто используемая активационная функция (рисунок 16).

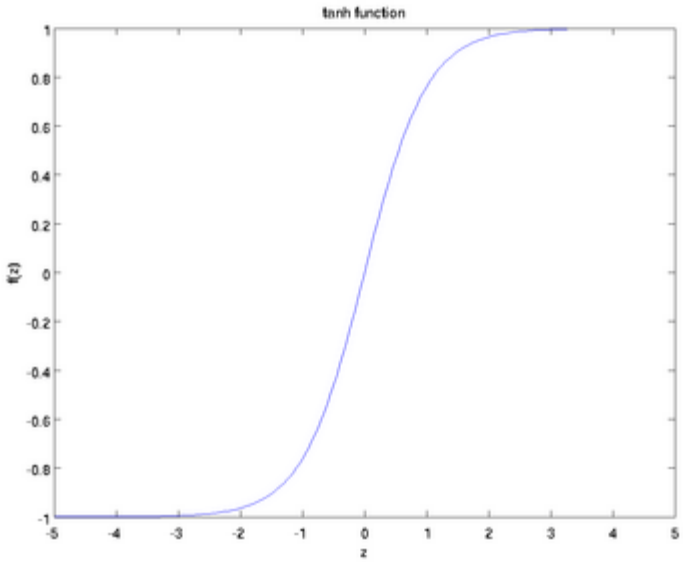


Рисунок 16 – функция гиперболического тангенса

Гиперболический тангенс очень похож на сигмоиду. И действительно, это скорректированная сигмоидная функция.

Поэтому такая функция имеет те же характеристики, что и у сигмоиды, рассмотренной ранее. Её природа она нелинейна, хорошо подходит для комбинации слоёв, а диапазон значений функции от (-1, 1). Поэтому нет смысла беспокоиться, что активационная функция перегрузится от больших значений. Однако стоит отметить, что градиент тангенциальной функции больше, чем у сигмоиды (производная круче). Решение о том, выбрать ли сигмоиду или тангенс, зависит от ваших требований к амплитуде градиента. Так же, как и сигмоиде, гиперболическому тангенсу свойственная проблема исчезновения градиента. Тангенс также является очень популярной и используемой активационной функцией.

Рассмотрим функцию активации Relu, рисунок 18. Пользуясь определением, становится понятно, что ReLu возвращает значение x, если x положительно, и 0 в противном случае. Схема работы приведена ниже.

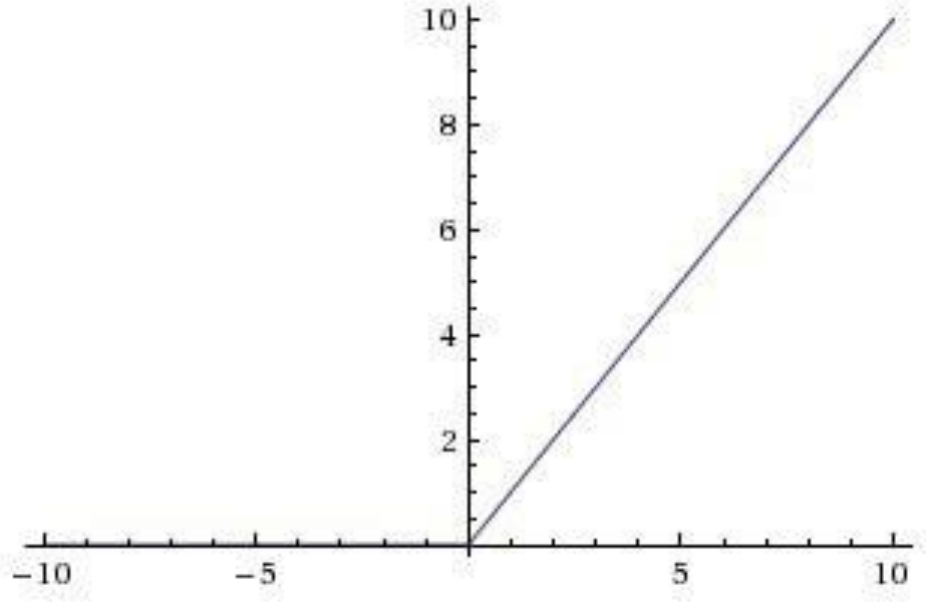


Рисунок 17 – Функция ReLu

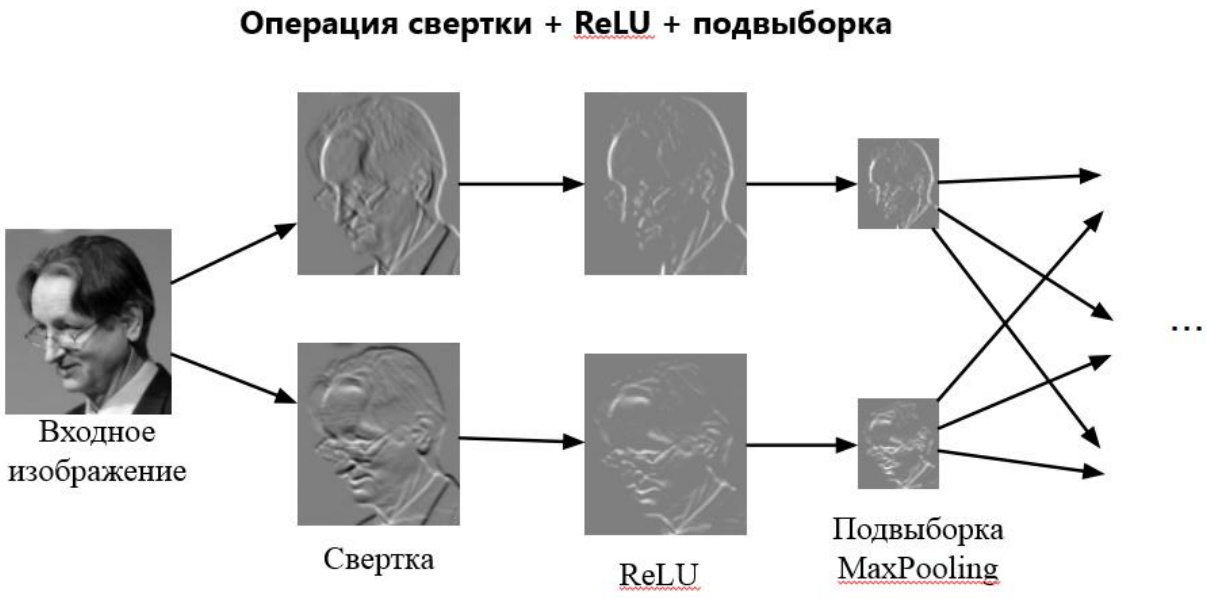


Рисунок 18 – Операция свертки + ReLu

На первый взгляд кажется, что ReLu имеет все те же проблемы, что и линейная функция, так как ReLu линейна в первом квадранте. Но на самом деле, ReLu нелинейна по своей природе и комбинация ReLu также нелинейна. На самом деле, такая функция является хорошим аппроксиматором, так как любая функция может быть аппроксимирована комбинацией ReLu. Это означает, что мы можем

проходить слои. Область допустимых значений ReLu – [0,inf), то есть активация может «взорваться»[15] .

Следующий пункт – разреженность активации. Представим большую нейронную сеть с множеством нейронов. Использование сигмоиды или гиперболического тангенса будет тянуть за собой активацию всех нейронов аналоговым способом. Это означает, что почти все активации должны быть обработаны для описания выхода сети. Другими словами, активация плотная, а это затратно. В идеале мы хотим, чтобы некоторые нейроны не были активированы, это сделало бы активации разреженными и эффективными.

ReLu позволяет это сделать. Представим сеть со случайно инициализированными весами (или нормализированными), в которой примерно 50 % активаций равны 0 из-за характеристик ReLu (возвращает 0 для отрицательных значений x). В такой сети включается меньшее количество нейронов (разреженная активация), а сама сеть становится легче. Отлично, кажется, что ReLu подходит нам по всем параметрам. Но ничто не безупречно, в том числе и ReLu. Из-за того, что часть ReLu представляет из себя горизонтальную линию (для отрицательных значений X), градиент на этой части равен 0. Из-за равенства нулю градиента, веса не будут корректироваться во время спуска. Это означает, что пребывающие в таком состоянии нейроны не будут реагировать на изменения в ошибке/входных данных (просто потому, что градиент равен нулю, ничего не будет меняться). Такое явление называется проблемой умирающего ReLu (Dying ReLu problem). Из-за этой проблемы некоторые нейроны просто выключатся и не будут отвечать, делая значительную часть нейросети пассивной. Однако существуют вариации ReLu, которые помогают эту проблему избежать. Например, имеет смысл заменить горизонтальную часть функции на линейную. Если выражение для линейной функции задается выражением $y = 0{,}01x$ для области $x<0$, линия слегка отклоняется от горизонтального положения. Существует и другие способы избежать нулевого градиента. Основная идея здесь – сделать градиент неравным нулю и постепенно восстанавливать его во время тренировки.

ReLu менее требовательно к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции. Поэтому имеет смысл использовать ReLu при создании глубоких нейронных сетей[16] .

Настало время решить, какую из функций активации использовать. Следует ли для каждого случая использовать ReLu? Или сигмоиду? Или tanh? На эти вопросы нельзя дать однозначного ответа. Когда вы знаете некоторые характеристики функции, которую пытаетесь аппроксимировать, выбирайте активационную функцию, которая аппроксимирует искомую функцию лучше и ведет к более быстрому обучению.

Например, сигмоида хорошо показывает себя в задачах классификации, так как аппроксимацию классифицирующей функции комбинацией сигмоид можно провести легче, чем используя ReLu, например.

Используйте функцию, с которой процесс обучения и сходимость будут быстрее. Более того, вы можете использовать собственную функцию. Если вы не знаете природу исследуемой функции, в таком случае начните с ReLu и потом работайте в обратном направлении. В большинстве случаев ReLu работает как хороший аппроксиматор.

2.5 Выводы по главе 2

По итогам проведенного анализа методов машинного обучения был выбран метод сверточных нейронных сетей. Метод сверточных нейронных сетей при обучении показывает более точные значения, процесс обучения и сходимость происходят быстрее.

По итогам анализа функции активации нейросети было выявлено, что при выборе функции нельзя дать однозначного ответа, необходимо использовать функцию в зависимости от задачи.

В целом можно сказать, что для решения задачи регрессии подходят множество алгоритмов, каждый алгоритм имеет свои достоинства и недостатки, связанные с

их спецификой. Деревья решений лучше всего работают с бинарными признаками, которые не нуждаются в масштабирование, а нейронные сети, наоборот, лучше работают с непрерывными признаками и нуждаются в масштабирование.

3 ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ЭМОЦИЙ

3.1 Исследование эффективности сверточных нейронных сетей на примере простой нейронной сети с описанием библиотек

Рассмотрим библиотеку Keras. Главное библиотекой на которой основывается данная работа является библиотека Keras – это одна из самых популярных библиотек для Python с открытым исходным кодом, которая позволяет легко создавать нейронные сети. Библиотека совместима с TensorFlow, Microsoft Cognitive Toolkit, Theano и MXNet и Theano являются наиболее часто используемыми численными платформами на Python для разработки алгоритмов глубокого обучения. Глубокое обучение – это метод машинного обучения. Глубокое обучение позволяет обучать модель предсказывать результат по набору входных данных. Для обучения сети можно использовать как контролируемое, так и неконтролируемое обучение.

Понять структуру нейронной сети проще на простом примере. Пример состоит из 3-х слоев:

- входной слой;
- скрытый слой (слои);
- выходной слой.

Входной слой принимает входные данные. В нашем случае имеется четыре нейрона на входном слое: аэропорт вылета, аэропорт назначения, дата вылета и авиакомпания. Входной уровень передает эти данные в первый скрытый слой.

Скрытые слои выполняют математические вычисления со входными данными. Одна из задач при создании нейронных сетей – определение количества скрытых слоев и нейронов на каждом слое.

Слово «глубина» в термине «глубокое обучение» означает наличие более чем одного скрытого слоя.

Выходной слой выдает результат. В нашем случае это прогноз цены на билет.

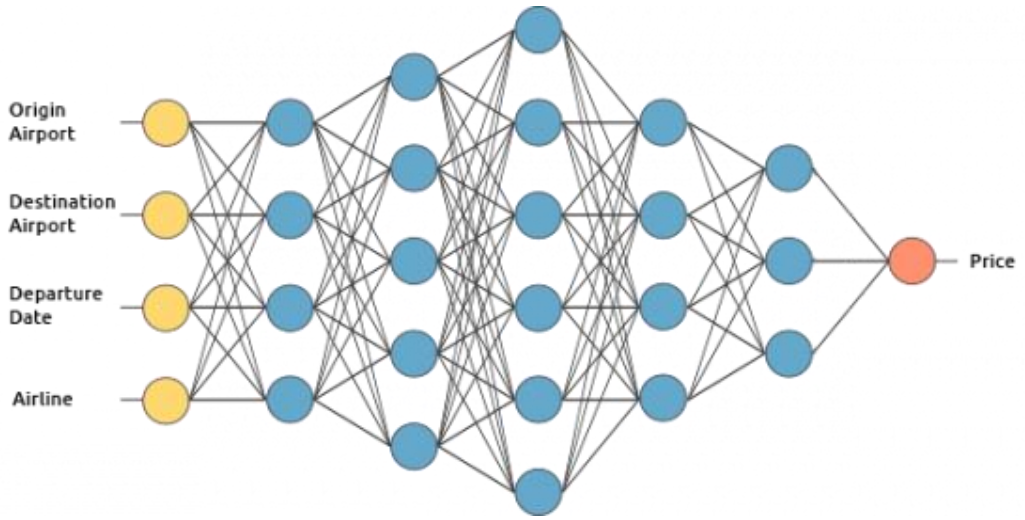


Рисунок 19 – Выходной слой с прайсом

Отсюда мы видим (рисунок 19), что данная работа нейроны связаны между собой с определенным весом. Вес определяет важность элемента входных данных. Исходные веса задаются случайным образом [17].

При прогнозировании цены на билет дата вылета является одним из наиболее важных факторов. Следовательно, связи нейрона времени вылета будут иметь большой вес (рисунок 20).

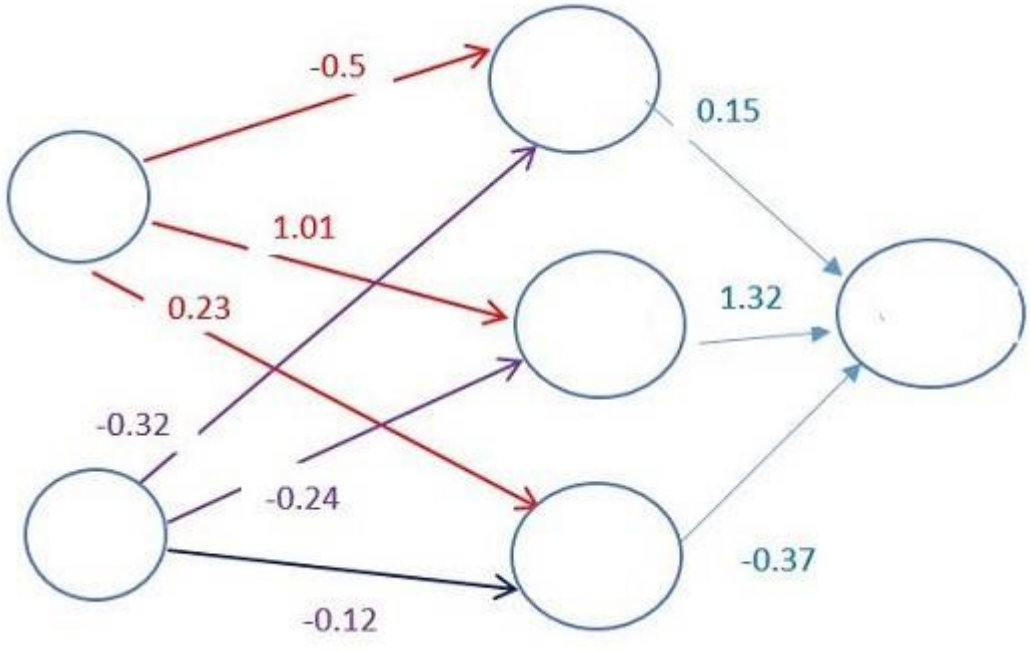


Рисунок 20 – Функции активации нейронов

Каждый нейрон имеет функцию активации их мы подробно разобрали в первой главе. Ее смысл трудно понять без привлечения математических рассуждений. Одной из ее целей является «стандартизация» данных на выходе из нейрона.

После того, как набор входных данных прошел через все слои нейронной сети, функция активации возвращает выходные результаты через выходной уровень. Обучение нейросети – самая сложная часть глубокого обучения нужен большой набор данных и нужно большое количество вычислительной мощности.

Для оценки стоимости билета нужно найти исторические данные о ценах на билеты. Из-за большого количества возможных комбинаций аэропортов и дат вылета нам нужен очень большой список цен на билеты.

Для обучения сети нужно подать в нее подготовленные данные и сравнить сгенерированные ей выходные результаты с результатами из нашего тестового набора данных. Поскольку сеть еще не обучена, результаты будут неверными.

После пропуска всех данных можно определить функцию, которая будет показывать нам, насколько результаты работы алгоритма отличаются от реальных данных. Эта функция называется функцией потерь.

В идеале мы хотим, чтобы функция потерь была равна нулю. В этом случае выходные результаты работы сети полностью совпадают с результатами тестового набора данных.

Еще одним важной библиотекой, на которой строится моя модель это TensorFlow – библиотека или фреймворк разработанный компанией Google, который предназначен для проектирования, создания и изучения моделей глубокого обучения. Глубокое обучение – это область машинного обучения, алгоритмы в которой были вдохновлены структурой и работой мозга. Вы можете использовать TensorFlow, чтобы производить численные вычисления. Само по себе это не кажется специфичным, однако эти вычисления производятся с помощью data-flow графов. В этих графах вершины представляют собой математические операции, в то время как ребра представляют собой данные, которые обычно представляются в виде многомерных массивов или тензоров, которые сообщаются

между этими ребрами. Чтобы хорошо понять тензоры, следует иметь хорошие знания в линейной алгебре и уметь производить вычисления с векторами. Тензоры реализованы в TensorFlow как многомерные массивы данных. Перед тем, как мы перейдем к плоским векторам, неплохо бы вспомнить о том, что такое вектор. Вектор – это особый вид матрицы, прямоугольный массив с числами. Так как векторы – это упорядоченный набор чисел, то они часто представляются в виде столбцов матриц. Другими словами, вектор – скалярная величина, которой дали направление**.** Длина математического вектора – величина абсолютная, в то время как направление – относительная. Длина измеряется относительно направления, а в качестве единиц выступают градусы или радианы. Обычно считается, что направление положительно и отсчитывается против часовой стрелки относительно начальной точки отсчета. Визуально, конечно, векторы представляют собой стрелки, как на картинке выше. Это означает, что вы можете рассматривать векторы, как стрелки определенной длины и направления. Плоский вектор – это простейший тензор. Они очень похожи на обычные векторы, такие как вы видели выше, с тем небольшим отличием, что они могут сами определять себя в векторном пространстве. Плоский вектор – это частный случай тензора. Помните, вектор определялся в прошлом разделе как скаляр, которому задали направление. Тензор же – это математическое представление физической сущности, которая может быть задана величиной и несколькими направлениями.[17]

Keras, наоборот, предоставляет простой и удобный способ создания моделей глубокого обучения. Ее создатель, François Chollet, разработал ее для того, чтобы максимально ускорить и упростить процесс создания нейронных сетей. Он сосредоточил свое внимание на расширяемости, модульности, минимализме и поддержке Python. Keras можно использовать с GPU и CPU, он поддерживает как Python 2, так и Python 3. Keras компании Google внесла большой вклад в коммерциализацию глубокого обучения и искусственного интеллекта, поскольку она содержит современные алгоритмы глубокого обучения, которые ранее были не только недоступными, но и непригодными для использования.[18]

С помощью анализа эмоций можно определить отношение (например, настроение) человека к тексту, взаимодействию или событию. Поэтому сентимент-анализ относится к области обработки естественного языка, в которой смысл текста должен быть расшифрован для извлечения из него тональности и настроений.

Спектр настроений обычно подразделяется на положительные, отрицательные и нейтральные категории. С использованием анализа настроений можно, например, прогнозировать мнение клиентов и их отношение к продукту на основе написанных ими обзоров. Поэтому анализ настроений широко применяется к обзорам, опросам, текстам и многому другому.

Рассмотрим библиотеку NumPy. Вторая одна из главных библиотек, NumPy – это фундаментальный пакет для научных расчетов с помощью Python. Это библиотека, предоставляющая такие объекты как многомерные массивы, различные производные объекты, а также набор процедур для быстрых операций с массивами, включая математические, логические, управление формой, сортировкой, выборкой, вводом/выводом, дискретными преобразованиями Фурье, простой линейной алгебры, простыми статистическими операциями, симуляцией случайностей и другие. В основе ядра пакета NumPy лежат объекты ndarray. Они могут содержать n-мерные массивы однородного типа данных со множеством возможных операций с помощью скомпилированного кода. Есть несколько важных различий между NumPy массивами и стандартными последовательностями языка Python.

Одной из проблем NumPy это то что массивы имеют фиксированный размер, заданный при создании, а списки в Python могут менять размер динамически. Изменение размера объекта ndarray просто создаст новый массив и удалит оригинальный. Элементы в массиве NumPy должны быть одного типа данных и занимать одинаковый размер в памяти. Исключением являются массивы объектов (как в Python так и в NumPy), у которых может быть разный размер. NumPy массивы облегчают сложные математические и другие операции над большими объемами данных. Обычно, такие операции выполняются более эффективно и

требуют меньше кода, чем при использовании обычных встроенных в Python возможностей по работе с последовательностями. Большое и постоянно растущее количество научных и математических пакетов на основе Python используют массивы NumPy. Хотя в качестве входных данных они берут обычные последовательности Python, затем они конвертируют их в массивы NumPy перед обработкой, и они выдают данные в виде NumPy массива. Другими словами, чтобы эффективно использовать множество (а может быть и большинство) научных и математических программ, основанных на Python – простох навыков работы со встроенными типами последовательностей Python становится малоэффективно – нужно также знать, как использовать NumPy массивы.

Моменты, связанные с размеров последовательностей и скоростью особенно важны в научных расчетах. В качестве простого примера можно рассмотреть умножение каждого элемента одной одномерной последовательности на соответствующий элемент другой последовательности такой же длины. Если данные бы хранились в двух Python списках (a и b), мы бы могли перебрать каждый элемент. Это выдаст нам правильный результат, но, если a и b содержать по миллионам элементов, мы дорог*о заплатим за неэффективность таких циклов в Python. Мы могли бы выполнить эту задачу гораздо быстрее в C таким кодом. Это сэкономит все накладные расходы, связанные с интерпретацией кода Python и управлением объектами, которые дают преимущества, связанные с самим программированием в Python. Кроме того, сложность кода растет при увеличении размерности наших данных. [19-20]

NumPy дает нам лучшее из обоих этих подходов: поэлементные операции – это стандартный режим работы, когда используются массивы ndarray, но такие операции очень быстро выполняются заранее скомпилированным C-кодом. А NumPy делает тоже, что и предыдущие примеры, но со скоростью почти равной языку C и с простотой кода, которую мы обычно ожидаем, основанного на Python. На самом деле подход NumPy даже проще.

Рассмотрим пакет pandas, который является наиболее важным инструментом для ученых и аналитиков, работающих сегодня на языке Python. Мощные инструменты машинного обучения, pandas является основой большинства проектов обработки данных.

У Pandas так много применений, что может иметь смысл перечислить то, что она не может сделать, вместо того, что она может сделать.

Этот инструмент, по сути, основа ваших данных. Через Pandas вы знакомитесь со своими данными, очищаете и анализируя их. Например, скажем, вы хотите изучить набор данных, хранящийся в CSV на вашем компьютере. Pandas будет извлекать данные из этого CSV в DataFrame – в основную таблицу – и затем позволит вам делать такие вещи. Очистите данные, выполнив такие действия, как удаление пропущенных значений и фильтрация строк или столбцов по некоторым критериям. Визуализируйте данные с помощью Matplotlib. Графики, линии, гистограммы, пузыри и многое другое. Сохраните очищенные, преобразованные данные обратно в CSV, другой файл или базу данных

Библиотека pandas не только является основным компонентом инструментария обработки данных, но и используется вместе с другими библиотеками [21].

Pandas построен поверх пакета NumPy, что означает, что большая часть структуры NumPy используется или копируется в Pandas. Данные в Pandas часто используются для статистического анализа, построения графиков функций из Matplotlib и алгоритмов машинного обучения в Scikit-learn.

Для простой работы с языком Python мы используем Jupyter Notebooks он предлагают хорошую среду для использования Pandas для исследования и моделирования данных.

Jupyter Notebooks дает нам возможность выполнять код в определенной ячейке, а не запускать весь файл. Это экономит много времени при работе с большими наборами данных и сложными преобразованиями. Jupyter также предоставляют простой способ визуализации панелей данных и графиков Pandas [22].

Рассмотрим библиотеку matplotlib.pyplot – это библиотека двумерной графики для языка программирования python с помощью которой можно создавать высококачественные рисунки различных форматов. Matplotlib представляет собой модуль-пакет для python. Данная библиотека умеет:

- графики (line plot);
- диаграммы рассеяния (scatter plot);
- столбчатые диаграммы (bar chart) и гистограммы (histogram);
- круговые диаграммы (pie chart);
- стволлист диаграммы (stem plot);
- контурные графики (contour plot);
- поля градиентов (quiver);
- спектральные диаграммы (spectrogram).

3.2 Построение моделей методом обучения нейронной сети

Нейронная сеть в данной работе построена из четырех сверточных слоев. В данном следовании количество сверточных слоев варьировалось от трех до пяти, что видно из таблицы 2 «изменение количества сверточных слоев», два слоя подвыборки maxpolling, слой преобразования данных Flatten, слой регуляризации dropout, и входной, и выходной полносвязанный слои danse. Архитектура сверточных нейронных сетей обеспечивает очень хорошую производительность для всех наборов данных, но сеть, использованная в этой работе, довольно проста, и именно это делает ее интересной для исследования и помогает понять основы Сверточных нейронных сетей. Несколько выдающихся результатов заключаются в том, что максимальный пул всегда превосходит средний пул, что идеальные размеры фильтров важны, но зависят от задачи, и что регуляризация, по-видимому, не сильно отличается от рассматриваемых задач НЛП. Предостережение этого исследования заключается в том, что все наборы данных были довольно схожи с точки зрения их длины документа, поэтому одни и те же рекомендации могут не применяться к данным, которые выглядят значительно различными [23].

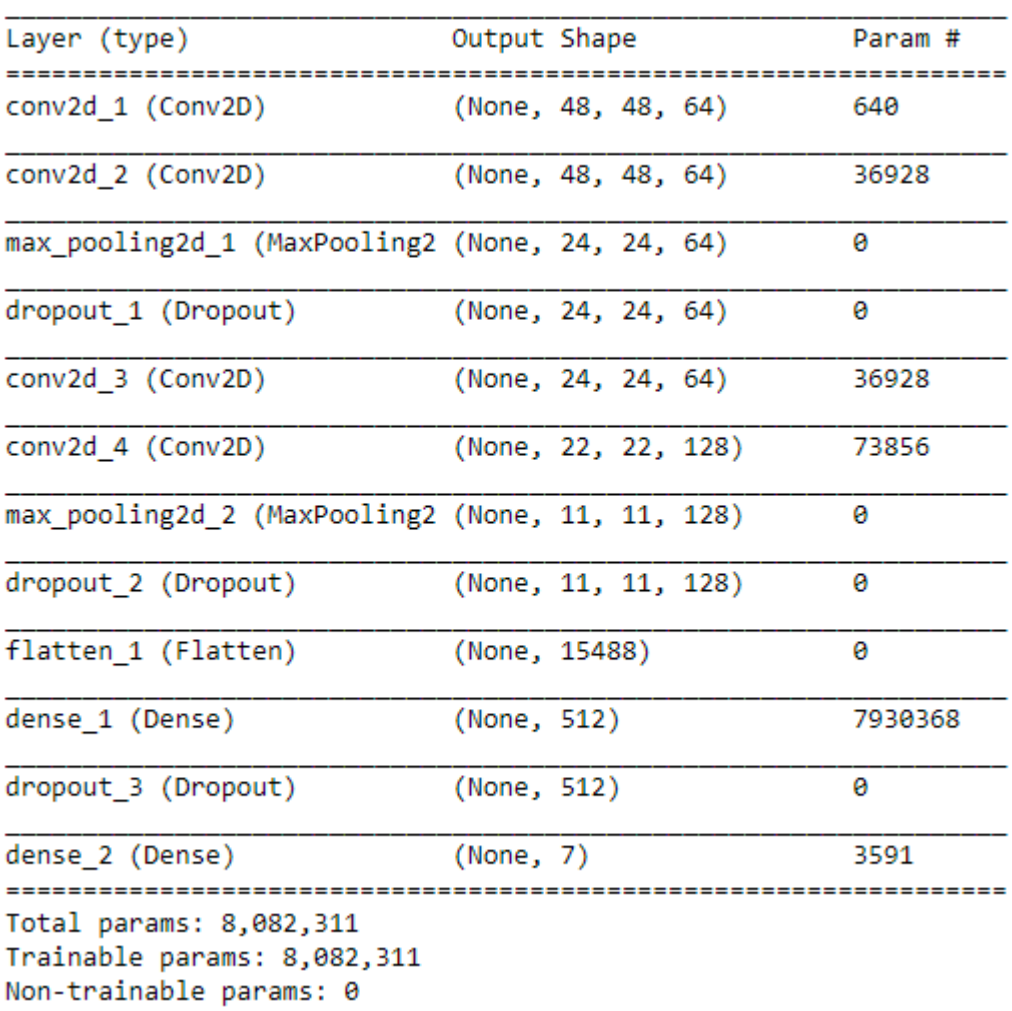


Рисунок 21 – Слои сверточных нейронной сети

В керасе необучаемые параметры означает количество весов, которые не обновляются во время тренировки с обратным распространением.

Есть в основном два типа не обучаемых весов:

– те, которые вы выбрали, чтобы сохранить постоянными при обучении. Это означает, что керасы вообще не будут обновлять эти веса во время тренировок;

– те, которые работают как статистика в слоях BatchNormalization. Они обновлены со средним и дисперсией, но они не "обучены обратному распространению".

Веса – это значения внутри сети, которые выполняют операции и могут быть скорректированы в соответствии с тем, что мы хотим. Алгоритм обратного распространения изменяет веса в сторону меньшей ошибки в конце [24].

3.3 Описание базы данных Feb2013

Набор данных Feb2013 применяется для обучения сверточных нейронных сетей для распознавания эмоций он содержит примерно 32 тысячи изображений. Данные состоят из полутоновых изображений лиц размером 48x48 пикселей. База данных была создан с помощью API поиска изображений Google лица были автоматически зарегистрированы, так что лицо более или менее отцентрировано и занимает примерно одинаковое количество места в каждом изображении. Задача состоит в том, чтобы классифицировать каждое лицо на основе эмоций, показанных в выражении лица, в одну из семи категорий (0 = агресия, 1 = отвращение, 2 = страх, 3 = радость, 4 = грусть, 5 = удивление, 6 = спокойствие).

Учебный набор состоит из 26709 штук изображений. А тестовый набор, используемый для проверки точности, состоит из 5589 штук изображений.

Набор данных Feb2013 был подготовлен Пьером-Люком Кэрриером и Аароном Курвиллем в рамках продолжающегося исследовательского проекта.

Рисунок 22 – Пример фотографий эмоции представленных в наборе Feb2013

3.4 Построение и сравнение моделей

Далее рассмотрены функции активации нейронной сети, из таблицы видно что совокупность функций relu и sigmoid, дает наилучший результат по сравнению с

остальными функциями активаций, как видно из рисунка 24, что точность обучения при любом количестве эпох выше.

Таблица 1 – Изменение функций активации

Функция активации	оптимизатор	Точность на обучающей выборке	Точность на тестовой выборке
Sigmoid sigmoid	adam	53	54
Relu sigmoid	adam	52,7	50,5
Relu softmax	adam	59,2	58,9
Sigmoid softmax	adam	42	42

На рисунке 23 видно, что максимальная точность достигается на значении примерно на 30 эпохе обучения, и в дальнейшем обучение практически прекращается.

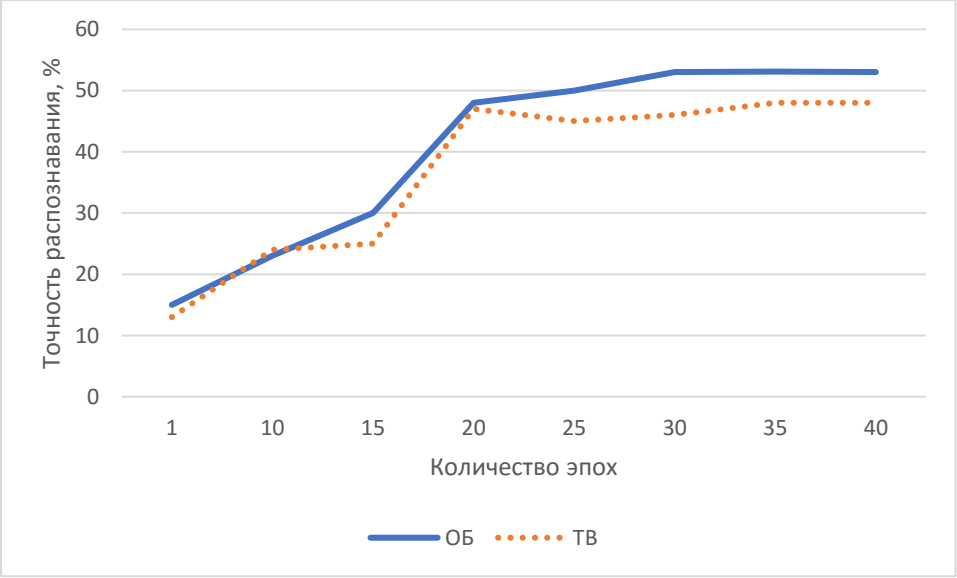


Рисунок 23 – Функция активации sigmoid

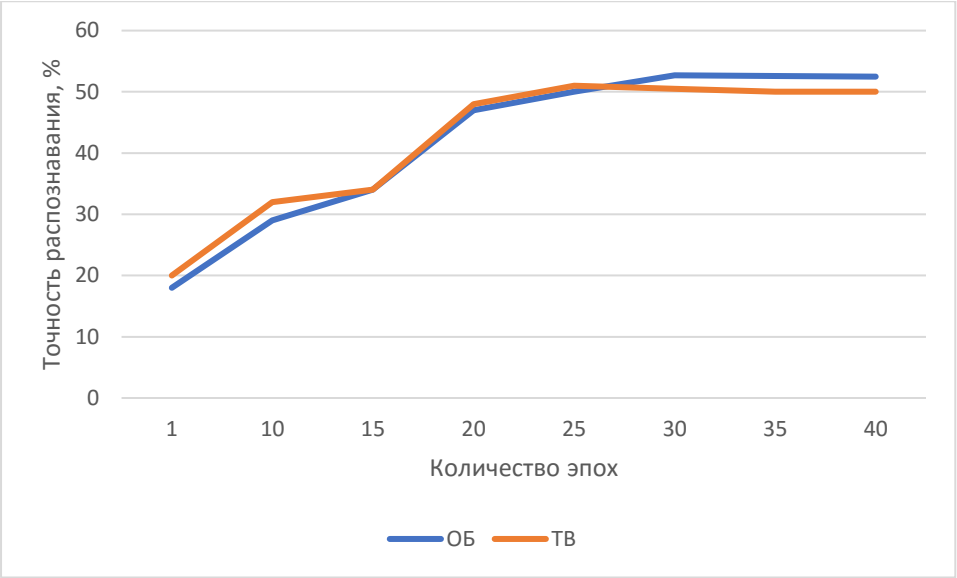


Рисунок 24 – Функции активациии Relu и sigmoid

На рисунке 25 можно видеть самую большую точность обучения как на обучающей, так и на тестовой выборке, данная точность достигнута с помощью функций активации relu и softmax.

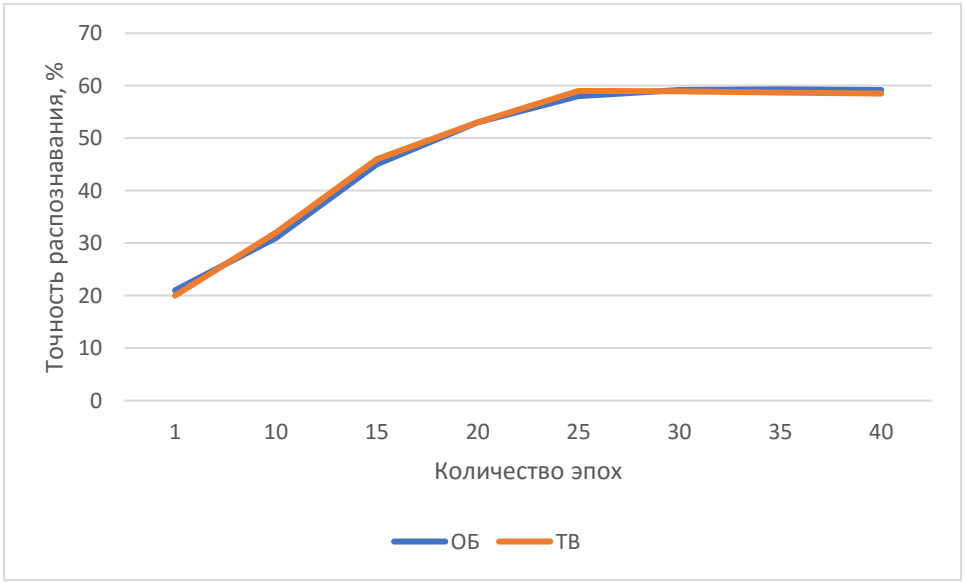


Рисунок 25 – Функция активации Relu и softmax

На рисунке 26 видно, что точность обучения низкая и это вязана с функцией активации sigmoid потому как данная функция лучше всего себя ведет с двоичной классификации и в моделях логистической регрессии.

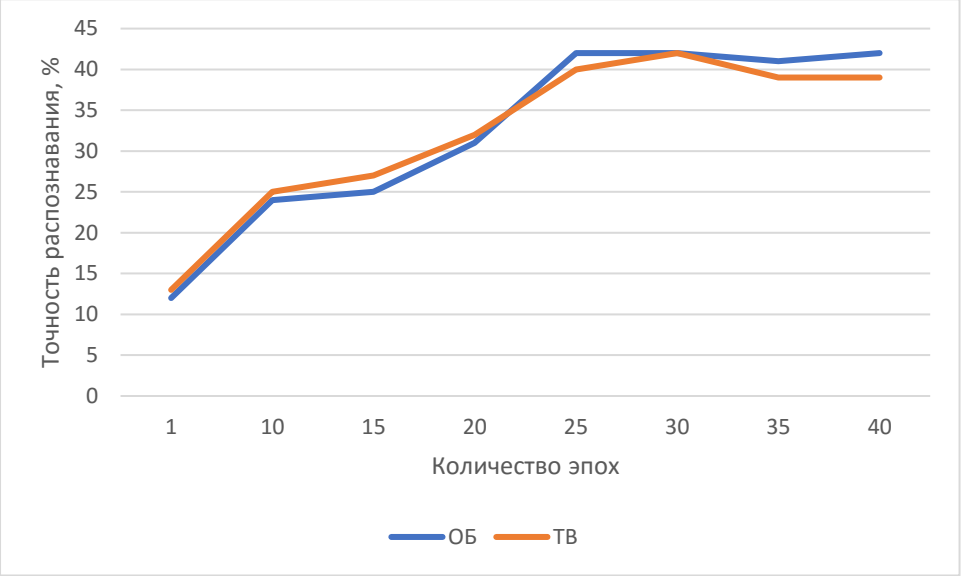


Рисунок 26 – Функция активации Sigmoid и softmax

Из таблицы 1 изменение количества сверточных слоев видно, что максимальная точность обучения при использовании функций активации rulu и sorfmax при четырех сверточных слоях.

Таблица 2 – Изменение количества сверточных слоев

	Точность на обучающей выборке	Точность на тестовой выборке
3	49,7	49,6
4	59,2	58,9
5	52.4	53

Из таблицы 2 видно, что изменение размера фильтра свёртки приводит к резкому уменьшению точности, а при 7x7 обучение вообще не происходит.

Таблица 3 – Изменение размерности фильтра

	Точность на обучающей выборке	Точность на тестовой выборке
1x1	23	23

3x3	59	58,9
5x5	25	24
7x7	Не обучается	Не обучается

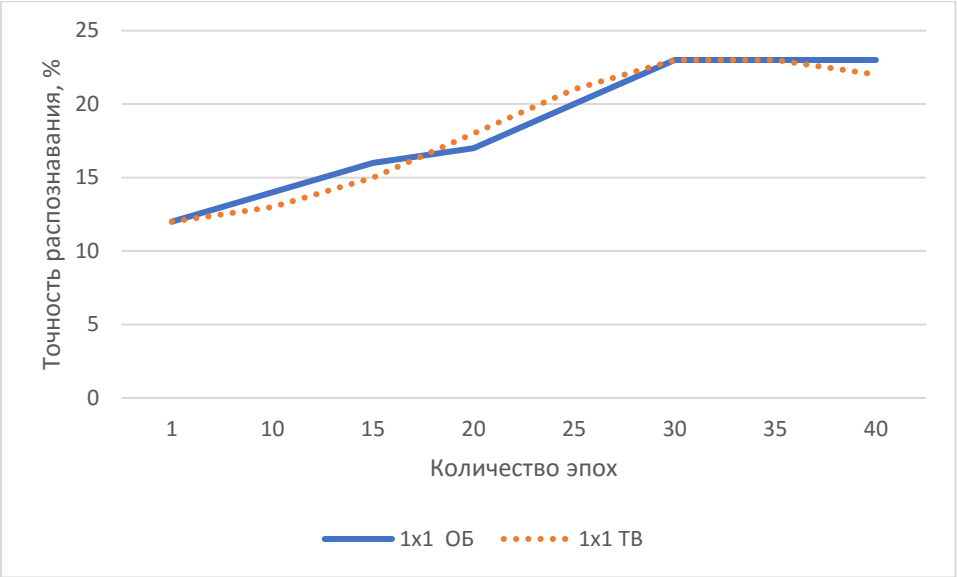


Рисунок 27 – Размер фильтра 1x1

Из рисунка 27 видно, что размер фильтра 1x1 обеспечивает точность около 23 % Свертка 1x1 – это специальный вид свертки, который интегрирует все каналы в одно значение, оставляя размер матрицы неизменным

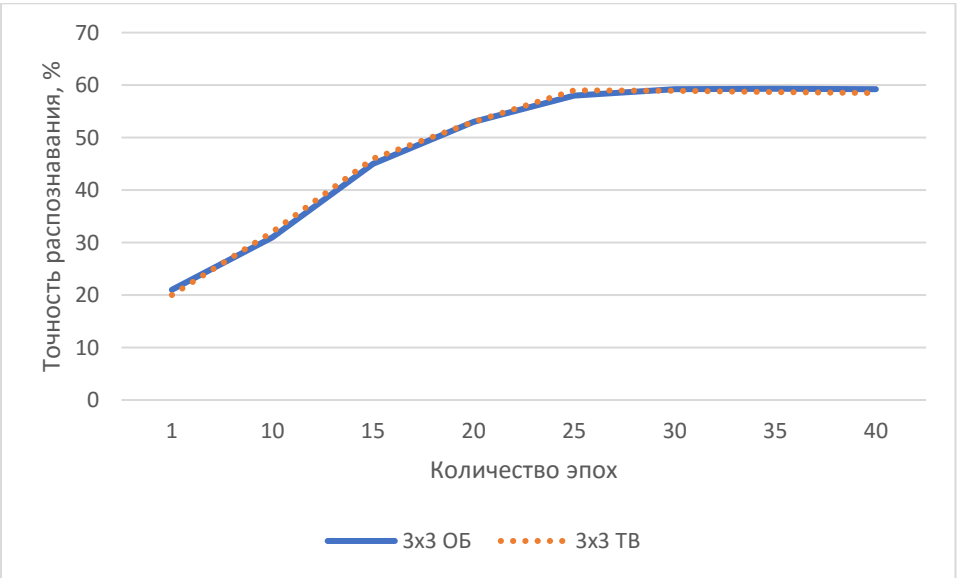


Рисунок 28 – Размер фильтра 3x3

Из рисунка 28 видно, что максимальная точность достигается при размере свертки 3x3, точность достигается до 59 %.

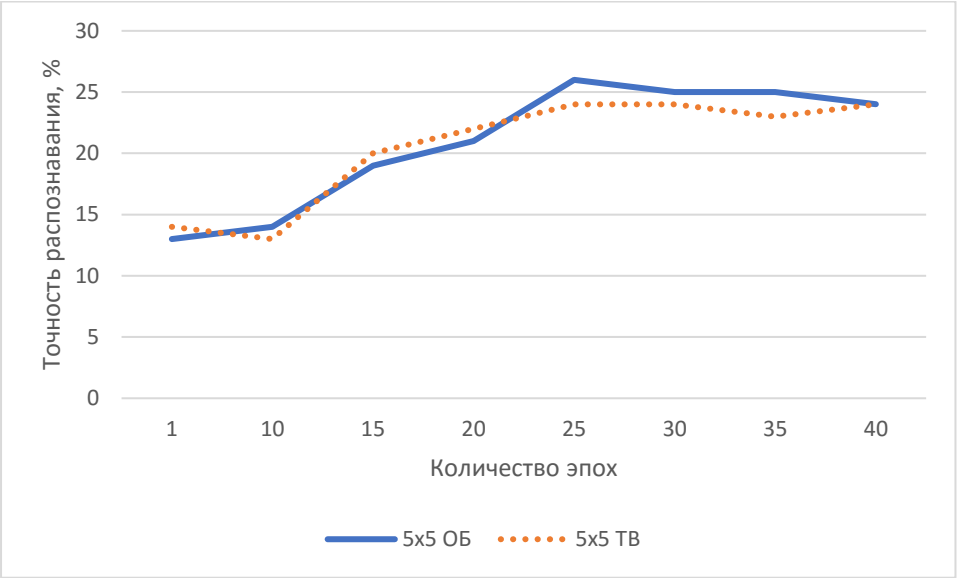


Рисунок 29 – Размер фильтра 5x5

Из рисунке 29 видно что нейронная сеть обучается только на 26 % при размере фильтра 5x5.

Таблица 4 – Изменение шага обучения

	Точность на обучающей выборке	Точность на тестовой выборке
0,3	42	42
0,4	45	44
0,5	59,2	58,9
1	Не обучается	Не обучается

Из таблицы 4 и рисунков 30–32 видно, что самая большая точность достигнута при размере шага 0,5. Длина шага – это параметр, который выбирается программистом. Высокая скорость обучения означает, что в обновлениях веса делались более крупные шаги, поэтому образцу может потребоваться меньше времени, чтобы набрать оптимальный набор весов. Но слишком высокая скорость

обучения может привести к очень крупным и недостаточно точным скачкам, которые помешают достижению оптимальных показателей.

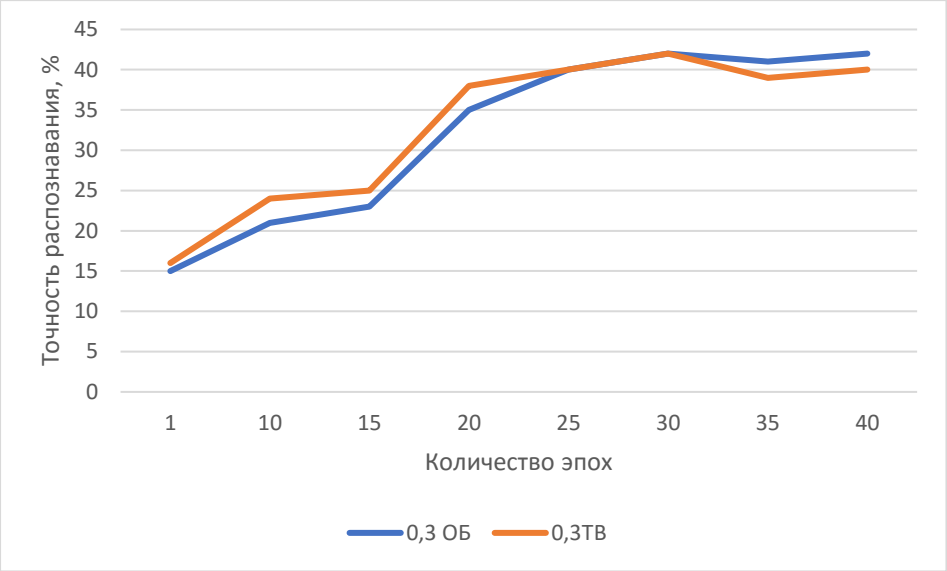


Рисунок 30 – Шаг обучение 0,3

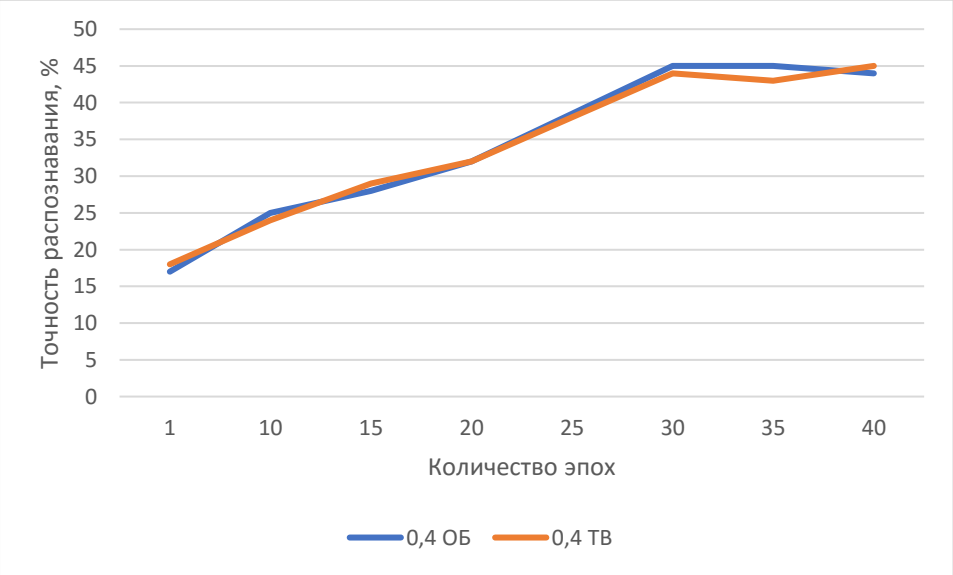


Рисунок 31 – Шаг обучение 0,4

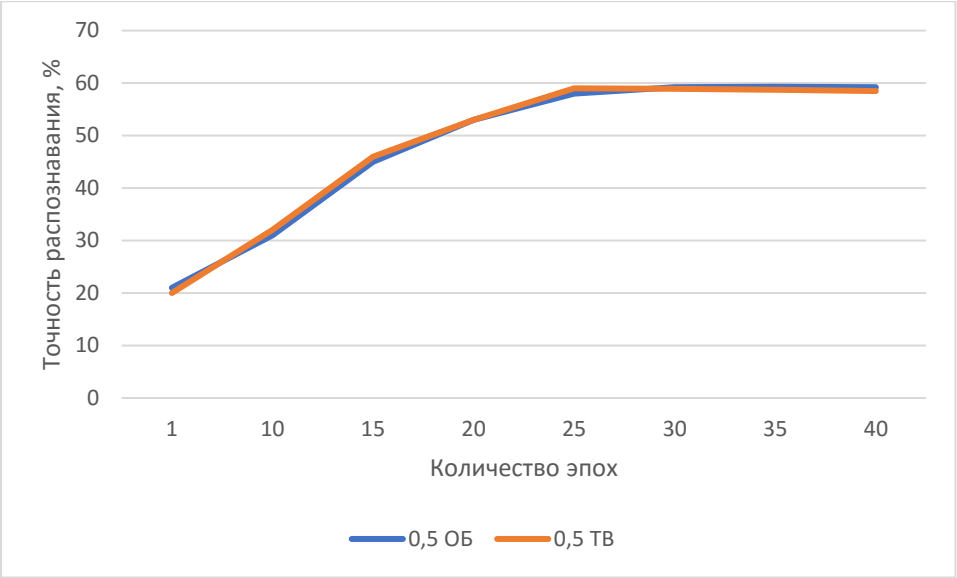


Рисунок 32 – Шаг обучения 0,5

Таблица 5 – Изменений эпох на максимальной полученной точной снс

Количество эпох и фолдов	Точность на обучающей выборке	Точность на тестовой выборке
20	52	53
30/5	59,2	58,9
40	59,3	58,8
50	Ошибка обучение. Из-за переобучения	

Из таблицы 5 видно, что 30 эпох обучения вполне достаточно, так как после 30 эпох обучение не происходит.

3.5 Метрика качества модели

Для оценки точности используется метрика Accuracy – это одна из популярных метрик качества моделей машинного обучения. Она часто используется для классификации и показывает долю данных, для которых класс был определен правильно. Хотя это не самая показательная метрика, но её проще всего считать, и она интуитивно понятная, поэтому применяется часто.

Наиболее очевидной мерой качества в задаче классификации является доля правильных ответов (accuracy).

$$\text{accuracy} = \frac{\sum_{i=1}^{\ell}[a(x_i) = y_i]}{\ell} \quad (1)$$

Данная метрика, имеет существенный недостаток. Если взять порог t меньше минимального значения прогноза b(x) на выборке или больше максимального значения, то доля правильных ответов будет равна доле положительных и отрицательных ответов соответственно. Таким образом, например, если в выборке 950 отрицательных и 50 положительных объектов, то при тривиальном пороге формула (1) t = maxi b(xi) мы получим долю правильных ответов 0,95. Это означает, что доля положительных ответов сама по себе не несет никакой информации о качестве работы алгоритма a(x), и вместе с ней следует анализировать соотношение классов в выборке. Также полезно вместе с долей правильных ответов вычислять базовую долю.

Кросс-валидация, которую иногда называют перекрестной проверкой, это техника валидации модели для проверки того, насколько успешно применяемый в модели метод способен работать на независимом наборе данных. Обычно кросс-валидация используется в ситуациях, где целью является предсказание, и хотелось бы оценить, насколько предсказывающая модель способна работать на практике. Один цикл кросс-валидации включает разбиение набора данных на части, затем построение модели на одной части (называемой тренировочным набором), и валидация модели на другой части (называемой тестовым набором). Чтобы уменьшить разброс результатов, разные циклы кросс-валидации проводятся на разных разбиениях, а результаты валидации усредняются по всем циклам.

3.6 Выводы по главе 3

При использовании алгоритма построение сверхточной нейронной сети, важно понимать достоинство и недостатки метода, учитывать природу данных с которым он лучше работает и объективно оценить работу алгоритма к масштабируемости

В главе были построены модели с использованием разных функций активации, размеров свертки, длины шага, размера фильтра, количества эпох и фолдов.

Исходя из этого, мы выявили, что для данной модели нейронной сети наиболее подходящие функции активации relu, softmax. Softmax можно использовать для любого количества классов. Он также используется для сотен и тысяч классов, например, в задачах распознавания объектов, где существуют сотни различных возможных объектов. Relu представляют собой специальную функцию, которая объединяет слои нелинейности и выпрямления в сверточных нейронных сетях.

4 РАЗРАБОТКА ПРИЛОЖЕНИЯ «МОНИТОРИНГ ЭМОЦИОНАЛЬНОЙ АКТИВНОСТИ ПОЛЬЗОВАТЕЛЕЙ»

4.1 Описание идеи создания приложения

При использовании каких-либо гаджетов детьми, родители могут не понимать, что они могут нанести вред психике ребенка, даже если он пользуется контентом с правильным возвратным рейтингом. Полностью отследить чем занимается ребенок можно с помощью детского аккаунта Google. Там можно увидеть сколько времени и во что ребенок играл на планшете, вести контроль за установленными приложениями, ограничивать по времени то или иное приложение, но все это не показывает реальную картину, все дети разные, кто-то любит одно, а кто-то другое. Дети способны найти такое видео или игровой контент, который их начнет злить, на протяжении определенного времени. И отследить это в реальном времени сложно если никто не приглядывает за ребенком. Целью данного модуля приложения будет анализ эмоции ребенка при просмотре видео или игр, в реальном времени, и идея такая что если ребенок начинает проявлять агрессию или гнев по отношению в данному контенту, то можно подавать сигнал родителям, приостанавливать приложение или блокировать экран устройства.

Вторым применением (модулем приложения) данной разработки может быть анализ эмоций кандидатов на собеседовании, например, использование отслеживания при прохождении психологических тестов, на профессиональную пригодность. Человек не способен долгое время контролировать себя настолько, чтобы отвечать на большое количество вопросов без эмоционально, отсюда можно выявить еще на этапе тестирования, насколько кандидат открыт и честен.

При проведении времени ребенком за экраном электронного устройства родителям всегда будет важно знать, насколько ребенок был вовлечен в процесс, насколько он был спокоен, испытывал положительные или отрицательные эмоции, и всегда ли он был зациклен на электронном устройстве (насколько абстрагировался от реального мира). Это важно в первую очередь, чтобы оценить

насколько ребенок зависим от того что происходит на экране, а также в какой степени контент влияет на его психику. Также оценка эмоционального состояния человека может быть полезна для работодателей в плане выявления кандидатов, которые не заинтересованы в вакансии, могут что-то скрывать, дают ложную или некорректную информацию, или отвлекаются на другие дела.

Идея приложения анализа эмоциональной активности пользователей заключается в мониторинге эмоций, сосредоточенности и заинтересованности. Фронтальная камера сканирует эмоции человека в режиме реального времени, пока пользователь увлечен происходящим на экране (Рисунок 33).

Рисунок 33 – пример метода сканирования эмоций у детей

Работа приложения будет построена из четырёх этапов:

1 этап:

В какой-то момент времени с определенной периодичностью (например, 1 раз в 30 секунд) будет происходить снимок с фронтальной камеры устройства и передаваться на обработку алгоритму локализации эмоций. Этот алгоритм сохраняет все обнаруженные эмоции во временную базу.

2 этап:

Все обнаруженные эмоции считываются из временной базы и обрабатываются методом распознавания эмоций – сверточными нейронными сетями.

3 этап:

Все успешно классифицированные эмоции вносятся в таблицу с указанием временной метки, когда происходило снятие кадра.

4 этап:

Производится сопоставление момента, в который человек испытал эмоцию с тем, что в этот момент было на экране гаджета (смартфона, планшетного компьютера, ноутбука). Количество успешно обнаруженных и распознанных эмоций сопоставляется с причиной такой эмоциональной реакции.

В дальнейшем программа может рассчитать количество эмоций, сопоставить причины эмоций. Также можно будет посмотреть статистику по каждому отдельно взятому человеку.

В данной программе будет полезной еще одна функция: что именно вызывает конкретные эмоции. С помощью этой функции можно будет автоматически отследить на какой контент у ребенка проявляются положительные или отрицательные эмоции.

4.2 План работ по разработке и коммерциализации приложения

Планируется создать приложение «Мониторинг эмоциональной активности пользователей» под каждую задачу будет отдельный модуль: контроль эмоционального состояния детей и контроль за активностью кандидатов, претендующих на вакантную должность.

Основные направления разработки приложений:

1) разработка и создание интерфейса программы;
2) исследование и реализация методов обработки видеопотока с веб-камеры устройства;
3) реализация метода распознавания эмоций на изображении;

4) разработка и создание приложения анализа эмоциональной активности пользователей устройства.

Рассмотрим, где приложение «Мониторинг эмоциональной активности пользователей» имеет практическую значимость:

1) использование частными лицами (родителям для отслеживания эмоционального состояния детей);

2) использование в коммерческих организациях при проведении собеседований с кандидатами на вакантную должность.

Рассмотрим подробнее сферы применения.

Модуль по распознаванию эмоций детей.

В век информационных технологий, гаджеты занимают прочное место в нашей жизни. Новое поколение детей не мыслит себя без электронных и цифровых устройств. Именно они в наше время облегчают людям жизнь, помогают в учебе и работе, и, конечно, ни один человек не обходится в современном мире без гаджетов. Насколько важны новые технологии в жизни ребенка, на это мы постараемся ответить Тема чрезвычайно актуальна у многих детей дошкольного возраста есть планшеты, и они с удовольствием ими пользуются. Но, необходимо четко расставить грани дозволенного во избежание психологических и социальных проблем со здоровьем у ребёнка.

В книге «Ребенок и гаджеты: психологическое исследование мнений современных родителей» [31] авторы Барсукова О. В., Мавлютова Е. В. и Савка М. А. выделяют основные правила использования гаджетов:

1)пример взрослого. Родителям рекомендуется в присутствии ребенка сократить время использования гаджетов;

2)ограничить ребенку время препровождения за любыми информационными носителями. (Не более 30 минут в день для детей 6–7 лет. Все что больше, уже вред здоровью ребенка);

3)контролировать контент, который ребенок просматривает. просматривать, с какого информационного источника идет просмотр или игра;

4)самостоятельно установить ребенку образовательные приложения (например, приложение для изучения цветов, детские математические приложения);

5) компьютерные игры, в которые играет ребенок, не должны заменять реальное общение с взрослым близким, друзьями;

6)свободная зона. Это территория, где гаджетам не место. Но зачастую, нам родителям, гаджеты приходят на выручку. Что приводит к проблемам. Из-за собственной лени, нехватки времени, нежелания, усталости после трудового дня, родители предоставляют своему ребенку планшет или смартфон.

Таким образом, родителям важно не допустить, чтобы гаджеты в жизни ребёнка стали источником проблем. Грамотное и умеренное их использование действительно будет способствовать развитию ребёнка и поможет ему шагать в ногу со временем.

Делая вывод, хочется отметить, что запрещать детям дошкольного возраста пользоваться современными устройствами не стоит. Сейчас, в век технологий, знания в области информационных технологий очень пригодятся в будущем. Главное, чтобы родители контролировали время, которое ребенок проводит за компьютером или планшетом, а также отслеживать просматриваемый детьми контент. Именно функцию контроля и будет осуществлять первый модуль приложения «Мониторинг эмоциональной активности пользователей».

Модуль по распознаванию эмоций кандидатов на вакантную должность.

Как показывает практика в области, при приеме на работу, а часто уже начав работать, выясняется, что кандидат/сотрудник не соответствует уровню занимаемой должности или требуемой квалификации. Стиль, манеры поведения, психологические свойства личности трудоустроенного также могут вызвать негативный отклик у работодателя, персонала компании.

А ведь оценить профессиональные компетенции, личностные качества работника можно еще до приема на работу, используя различные методики

интервью, а также тестирование. Тесты могут быть не только на бумажных носителях, но и в онлайн режиме с использованием электронных устройств.

Это помогает получать подробные, достоверные данные о кандидатах, увидеть полную и ясную картину, иллюстрирующую их профессиональные возможности с точки зрения психологических особенностей личности. При этом правильно подобранные методика позволяет значительно повысить качество результата, получаемого в ходе тестовых процедур.

Тестирование – наиболее распространенная форма оценки интеллектуального потенциала человека. Психодиагностические методики помогают выявить различные компоненты мышления, но ни в коем случае не поставить диагноз «умный – не умный». А. Анастази [32], классик психологической науки, отмечает, что «тесты интеллекта, как и другие тесты, следует использовать не для навешивания ярлыков на людей, а для их лучшего понимания». Хорошо, если такие методики в компании применяет психолог-профессионал, много времени посвятивший изучению тонкостей психодиагностики. В век информационных технологий стало возможным отследить эмоции кандидата при прохождении теста онлайн. Отследить как меняются эмоции человека, отвечающего на тесты можно с помощью встроенной веб-камеры в режиме реального времени. Именно это позволит сделать разрабатываемый модуль по распознаванию эмоций кандидатов на вакантную должность.

Применяя на предприятиях аттестационные методики, основанные в том числе и на тестировании, а также на анализе эмоций, которые испытывал кандидат во время тестирования, можно добиться больших показателей производительности труда, роста интеллектуального и профессионального уровня сотрудников, а, следовательно, повышения общей результативности работы компании, что обязательно уменьшит текучесть кадров.

4.3 Дорожная карта коммерциализации

Рассмотрим дорожную карту коммерциализации проекта (таблица 6).

Таблица 6 – Дорожная карта коммерциализации проекта на 2020 год

	2020			
	I квартал	II квартал	III квартал	IV квартал
Исследования и разработки	Исследование методов распознавания изображений. Исследование эффективности и точности распознавания эмоций.	Исследование методов распознавания изображений. Исследование эффективности и точности распознавания эмоций.	Анализ эффективности новых методов распознавания эмоций в онлайн режиме, разработка мер по улучшению производительности алгоритмов.	Тонкая настройка методов обучения сверточных нейронных сетей.
Создание продукта	Разработка и создание интерфейса приложения Создание функционала.	Разработка прототипа приложения «Мониторинг эмоциональной активности пользователей». Проверка приложения. Выявление слабых мест.	Тестирование прототипа приложения «Мониторинг эмоциональной активности пользователей». Публикация приложения в известных интернет-магазинах. Проверка и тестирование приложения, Исправление ошибок.	Добавление новых модулей в приложение. Расширение функционала приложения «Мониторинг эмоциональной активности пользователей».
Защита интеллектуальной собственности и лицензирование		Подача заявки на регистрацию прав собственности на новый продукт.		Лицензирование приложения «Мониторинг эмоциональной активности пользователей».

Окончание таблицы 6

Маркетинг, внедрение, продвижение	Исследование рынка аналогичных приложений.	Представление новых возможностей приложения на выставках и конференциях.	Представление новых возможностей приложения на конкурсах стартапов, реклама в соц. сетях Продвижение продукта за счёт рекламы в Интернете.	Продвижение продукта за счёт адресной рекламы в Интернете. Отладка взаимодействия с заказчиком. Продвижение сервиса «Мониторинг активности слушателей учебных курсов» в интернете.
Привлечение инвестиций и продажи		Привлечение инвестиций за счет конкурсов перспективных стартапов.	Демонстрация работы приложения «Мониторинг эмоциональной активности пользователей».	Продажа приложения не менее 50 пользователям.

4.4 Выводы по главе 4

В целях коммерциализации использования предложенного метода распознавания эмоций предлагается разработка и внедрение приложения «Мониторинг эмоциональной активности пользователей».

Приложение должно давать объективную информацию об эмоциональной активности в удобной для пользователя форме (графики, таблицы, схемы). В приложении можно будет не только ознакомиться с эмоциями, которые человек испытывает во время использования гаджета, но и получить информацию о том, какой именно контент стал их источником.

В главе подробно описаны требования, предъявляемые к приложению, основные этапы работы приложения и основные направления разработки, коммерциализация. Были описаны сферы применения и представлена дорожная карта проекта.

Таким образом, приложение «Мониторинг эмоциональной активности пользователей» имеет большую практическую значимость во многих сферах деятельности, является рентабельным для привлечения инвестиций и предполагает приносить прибыль от реализации.

ЗАКЛЮЧЕНИЕ

Работа направлена на исследование машинного обучения путем разработки построение моделей сверточной нейронной сети.

Так же стоит отметить, что в работе выявлены тенденции, приоритеты и направления в соответствии с методами машинного обучения, а также построены модели. В работе рассмотрены точки зрения различных авторов относительно понятия машинного обучения, компьютерного зрения и сверточных нейронных сетей.

Были проведены исследования методов машинного обучения и задач компьютерного зрения. После сопоставления результатов (тенденций) методов был сделан вывод о том, что для данной модели нейронной сети наиболее подходящие функции активации relu, softmax, размеры свертки 3x3, длины шага 0,5, 30 эпох и 5 фолдов.

Тем не менее, это только первый шаг в проделанной работе, и для получения более конкретных результатов, несомненно, нужны дополнительные исследования, для развития работы приложения «Мониторинг эмоциональной активности пользователей». Рассмотрены основные этапы работы приложения и основные направления разработки и коммерциализации приложения. Были описаны сферы применения и представлена дорожная карта проекта.

Таким образом, в данной работе были выполнены все поставленные задачи и цель магистерской диссертации была достигнута.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) L. Bottou O. Scaling Learning Algorithms towards AI To appear in “Large-Scale Kernel Machines”, / Chapelle, D. DeCoste, J.Weston (eds) MIT Press, 2007

2) Yann LeCun, Optimal Brain Damage, in Touretzky, David (Eds), Advances in Neural Information Processing Systems 2 (NIPS*89) / J. S. Denker, S. Solla, R. E. Howard and L. D. Jackel, Morgan Kaufman, Denver, CO, 1990

3) Y. LeCun, Y. Bengio: Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A. (Eds), The Handbook of Brain Theory and Neural Networks, MIT Press, 1995

4) Y. LeCun, L. Bottou, G. Orr and K. Muller: Efficient BackProp, in Orr, G. and Muller K. (Eds), Neural Networks: Tricks of the trade, Springer, 1998

5) Ranzato Marc'Aurelio, Christopher Poultney, Sumit Chopra and Yann LeCun: Efficient Learning of Sparse Representations with an Energy-Based Model, in J. Platt et al. (Eds), Advances in Neural Information Processing Systems (NIPS 2006), MIT Press, 2006

6) Галушкин А.И. Нейрокомпьютеры для обработки изображений [Текст] / А.И. Галушкин, Н.С. Томашевич, Е.И. Рябцев // Нейрокомпьютеры в прикладных задачах обработки изображений. Кн. 25. 2007. С. 74–109.

7) Krizhevsky A. ImageNet classification with deep convolutional neural networks [Text] / A. Krizhevsky, I. Sutskever, G.E. Hinton // Advances in Neural Information Processing Systems 25, 2012. Pp. 1106-1114.

8) Bengio Y. Learning deep architectures for AI [Text] / Y. Bengio // Foundations and Trends in Machine Learning. Vol. 2. Issue 1, 2009. Pp. 1-127.

9) Bengio Y. Scaling learning algorithms towards AI [Text] / Y. Bengio, Y. LeCun // in Large Scale Kernel Machines, MIT Press, 2007.

10) Le Q. Tiled convolutional neural networks [Text] / Q. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, A.Y. Ng // NIPS, 2010.

11) Ranzato M.A. What is the best multi-stage architecture for object recognition? [Text] / M.A. Ranzato, K. Jarrett, K. Kavukcuoglu, Y. LeCun// In ICCV, 2009.

12) Соколов Е.Н. Нейроинтеллект: от нейрона к нейрокомпьютеру [Текст] / Е.Н. Соколов, Г.Г. Вайткявичюс // М.: Наука, 1989. 238 с.

13) Шевелёв И.А. Нейроны-детекторы зрительной коры [Текст] / И.А. Шевелёв // М.: Наука, 2010. 183 с.

14) Eysel U.T. Pharmacological studies on receptive field architecture [Text] / U.T. Eysel // N.Y.: Acad. Press, 2002. Pp. 427-470.

15) LeCun Y. Backpropagation applied to handwritten zip code recognition [Text] / Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel // Neural Computation. Vol. 1, № 4, 1989. Pp. 541-551.

16) Хайкин С. Нейронные сети: полный курс, 2-е издание. Пер. с англ. [Текст] / С. Хайкин // М.: Издательский дом «Вильямс», 2008. 1104 с.: ил. Парал. тит. англ.

17) Гонсалес Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное [Текст] / Р. Гонсалес, Р. Вудс // М.: Техносфера, 2012. 1104 с.

18) Kavukcuoglu K. Learning Feature Hierarchies for Object Recognition [Text] / K. Kavukcuoglu // PhD diss. New York University, January 2011.

19) Kavukcuoglu K. Learning invariant features through topographic filter maps [Text] / K. Kavukcuoglu, K. Ranzato, M. Fergus, Y. LeCun // In CVPR’09, IEEE, 2009.

20) Larochelle H. An empirical evaluation of deep architectures on problems with many factors of variation [Text] / H. Larochelle, D. Erhan, A. Courville, J. Bergstra, Y. Bengio // In Twenty-fourth International Conference on Machine Learning (ICML, 2007).

21) Neal R.M. Connectionist learning of belief networks [Text] / R.M. Neal // PhD thesis, Department of Computer Science. University of Toronto, 1994.

22) Ackley D.H. A learning algorithm for Boltzmann machines [Text] / D.H. Ackley, G.E. Hinton, T.J. Sejnowski // Cognitive Science. Vol. 9, 1985. Pp. 147-169.

23) Le Q. Building high-level features using large scale unsupervised learning [Text] / Q. Le, A. Ranzanto, R. Monga, M. Devin, G. Corrado, K. Chen, J. Dean // Acoustics, Speech and Signal Processing (ICASSP), 2013.

24) Farabet C. Hardware accelerated convolutional neural networks for synthetic vision systems [Text] / C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, E. Culurciello // In International Symposium in Circuits and Systems (ISCAS'10). IEEE. Paris, May 2010.

25) Ciresan D. Multicolumn Deep Neural Networks for Image Classification [Text] / D. Ciresan, U. Meier, J. Schmidhuber // In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR '12. Pp. 3642-3649. Washington. DC. USA, 2012. IEEE Computer Society.

26) Червяков Н.И. Нейрокомпьютеры в остаточных классах. Кн. 11. [Текст] / Н.И. Червяков, П.А. Сахнюк, А.В. Шапошников, А.Н. Макоха // М.: Радиотехника, 2003. 272 с.

27) Влияние планшета на ребенка (2–13 лет) [Электронный ресурс] // Информационный портал https://geektimes.ru

28) Ерошкина М. Как бороться с детской телеманией? [Электронный ресурс] // «ШколаЖизни.ру» – ежедневный познавательный журнал – режим доступа: www.xpomo.com (дата обращения: 06.05.2017г.).

29) Колиснык В. «Телевизор калечит детей» // Вечерние Вести. 2001. – № 080 (576). Медведева И., Шишова Т. «Телеэксперимент над детьми» [Электронный ресурс] // Сайт Интернет против телеэкрана – режим доступа: www.contr-tv.ru (дата обращения 12.03.2019)

30) Пацлаф Р. «Застывший взгляд. Физиологическое воздействие телевидения на развитие детей». Источник: Rainer Patzlaff «Der gefrorene Blick. Physiologische Wirkungen des Fernsehens und die Entwicklung des Kindes».

31) Барсукова О. В., Мавлютова Е. В., Савка М. А. Ребенок и гаджеты: психологическое исследование мнений современных родителей // Вопросы дошкольной педагогики. – 2016. – №1. – С. 14-18.

32) Психологическое тестирование. Анастази А., Урбина С. 7-е изд. - СПб.: 2007. - 688 с

33) Сверточные нейронные сети с нуля [электронный ресурс] – Режим доступа. – URL: https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1 (дата обращения 01.06.2019).

34) Распознавание выражений лица и анализ ложных срабатываний между классами с использованием CNN [электронный ресурс] – Режим доступа. – URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062920073&doi=10.1007%2f978-3-030-12385-7_5&partnerID=40&md5=313947e528407eeba4eeb6c8c1eeae24 (дата обращения 22.05.2019).