

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им. В.П.Астафьева»

ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И ИНФОРМАТИКИ
Кафедра информатики и информационных технологий в образовании

Тихонов Алексей Анатольевич

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**ВВЕДЕНИЕ В МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ОБУЧАЮЩИХСЯ 10-11
КЛАССОВ**

Направление подготовки 44.04.01 Педагогическое образование
Направленность (профиль) образовательной программы
Информационные и суперкомпьютерные технологии в математическом
образовании

ДОПУСКАЮ К ЗАЩИТЕ:

Заведующий кафедрой

проф., д-р пед. наук,

Пак Н.И.

26.12.2020г.

(дата, подпись)

Руководитель магистерской программы

проф., д-р пед. наук, канд. физ.-мат. наук

Майер В.Р.

26.12.2020г.

(дата, подпись)

Научный руководитель

канд. физ.-мат. наук, Романов Д.В.

26.12.2020г.

(дата, подпись)

Обучающийся

Тихонов А.А.

26.12.2020г.

(дата, подпись)



Красноярск 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ4

Раздел 1: БОЛЬШИЕ ДАННЫЕ (Big Data) И ГЛУБОКОЕ МАШИННОЕ ОБУЧЕНИЕ (Deep Learning) В НЕЙРОННЫХ СЕТЯХ6

- 1.1. Понятие больших данных (Big Data)6
- 1.2. Сферы использования больших данных6
- 1.3. Принцип работы больших данных7
- 1.4. Методики анализа больших данных7
- 1.5. Аналитический инструментарий11
- 1.6. Ведущие поставщики13
- 1.7. Примеры использования больших данных14
- 1.8. Понятия Deep Learning и нейронная сеть20
- 1.9. Искусственные нейроны и их объединение в сеть21
- 1.10. Архитектура нейронных сетей23
- 1.11. Способы обучения нейронных сетей26
- 1.12. Сферы использования глубинного обучения в нейронных сетях34
- 1.13. Перспективы по модернизации нейронных сетей36
- 1.14. Применение нейронных сетей в сфере обучения38

Раздел 2: ПРАКТИЧЕСКАЯ РАБОТА ПО ОБУЧЕНИЮ НЕЙРОННОЙ СЕТИ40

- 2.1. Задача и необходимые инструменты для практического выполнения глубинного обучения нейронной сети 140
- 2.2. Задача и необходимые инструменты для практического выполнения глубинного обучения нейронной сети 250

Раздел 3: ЭЛЕКТИВНЫЙ КУРС66

- 3.1. Рабочая программа элективного курса БОЛЬШИЕ ДАННЫЕ (Big Data) И ГЛУБОКОЕ МАШИННОЕ ОБУЧЕНИЕ (Deep Learning) В НЕЙРОННЫХ СЕТЯХ.66
- 3.2. Пояснительная записка66
- 3.3. Тематическое планирование курса70
- 3.4. Содержание курса72
- 3.5. Выявление уровня освоения содержания74
- 3.6. Контрольные мероприятия75
- 3.7. Обоснование системы оценивания77
- 3.8. Учебно-методическое обеспечение78

ЗАКЛЮЧЕНИЕ81

ИСТОЧНИКИ И ЛИТЕРАТУРА84

ВВЕДЕНИЕ

Актуальность темы исследования обусловлена необходимостью устранить недостаток информации в существующих учебных изданиях по направлениям Большие данные и Глубокое машинное обучение.

Существует огромная потребность на рынке труда в специалистах из данной области информационных технологий. При этом наблюдается дефицит информации даже в сети Интернет, зачастую, это отдельные лекции и презентации без системного подхода к процессу обучению, раскрывающие отдельные аспекты вышеупомянутых направлений.

Проблема исследования заключается в отсутствии обучающих материалов для школьников и студентов в современных изданиях в области информатики по направлениям Большие данные и Глубокое машинное обучение.

Объект исследования: обучение информатике в старших классах общеобразовательной школы.

Предмет исследования: технология глубокого машинного обучения (Deep Learning) в нейронных сетях на стыке с технологией Больших данных. В соответствии с актуальностью, объектом, предметом были намечены следующие **задачи исследования:**

- систематизировать и обобщить информацию по технологии глубокого машинного обучения и большими данными;
- изучить основы теории разработки и построения элективных курсов;
- разработать элективный курс по направлению, наполнение занятий, систему упражнений, а так же методы для текущего, промежуточного и итогового контроля на обучаемыми при прохождении элективного курса.

Цель исследования: создание элективного курса в области информатики «Введение в машинное обучение для обучающихся 10-11 классов» для учащихся старшей средней общеобразовательной школы и студентов.

Практическая значимость работы заключается в том, что программа элективного курса может быть применена преподавателями школ, а также полезна для студентов – будущих преподавателей информатики.

Методы исследования. В выпускной квалификационной работе применяются такие общенаучные методы исследования, как наблюдение, измерение, описание, сравнение, анализ, синтез, индукция, дедукция, аналогия и некоторые другие.

Эмпирическую базу исследования составили результаты анализа и практическая работа по машинному обучению.

Структура исследования. Выпускная квалификационная работа включает в себя введение, 3 раздела и 24 подраздела в них, в которых решаются поставленные исследовательские задачи, заключение, список источников и литературы. В первом разделе отражена теория о Big Data и Deep Learning, второй раздел содержит практическую часть – создание нейронной сети и ей обучение, в третьем разделе описан элективный курс, который опирается на первые два раздела.

Раздел 1: БОЛЬШИЕ ДАННЫЕ (Big Data) И ГЛУБОКОЕ МАШИННОЕ ОБУЧЕНИЕ (Deep Learning) В НЕЙРОННЫХ СЕТЯХ

1.1. Понятие больших данных (Big Data)

Под большими данными (англ. Big Data) подразумевают технологии обработки структурированных и неструктурированных данных, которые имеют постоянный прирост с целью извлечения скрытых закономерностей. В последнее время сформировалось отличие между обычными структурированными базами данных с чёткими и понятными обработками и большими данными, которые могут не иметь большой объём информации на начальном этапе, но обладают главной отличительной чертой – возможность приспособить их под задачу в результате использования различных методов «машинного обучения». Вообще определяющими характеристиками больших данных являются объём, вариативность, скорость и ценность [1].

Большой объём данных (от 100 Тбайт) позволяет более точно находить различные связи для дальнейшего представления аналитики в агрегированном, понятном для оператора виде. В современных условиях объём информации может достигать сотен петабайт и даже эксабайт [1].

Вариативность данных позволяет выявить зависимости там, где на первый взгляд их не стоит искать. Например, зависимость активности покупателей от погоды или зависимость продолжительности сна от потребления лекарств.

Скорость обработки информации в больших данных близка к реальному времени.

1.2. Сферы использования больших данных

Широкое распространение использования больших данных получили Интернет-ресурсы, занимающиеся поиском, банковской и финансовой аналитикой, рекламой, продажей товаров и услуг. Прежде всего, это вызвано возможностью получения большого объёма информации о действиях

пользователей на этих ресурсах. Рекомендованные к просмотру фильмы, к прослушиванию музыкальные композиции, к чтению книги и статьи, контекстная реклама и СПАМ-фильтры в почте, сайты знакомств [2, 3].

Данная технология использоваться в медицине, банковской и военной сферах, полиции, экономике государств, изучении социальных явлений и для управления большими и малыми социальными группами. На мой взгляд, большие данные являются относительно новым витком в информационных технологиях и в ближайшее время проникнут во все сферы деятельности человека, вытесняя обычные базы данных с их бизнес-аналитиками.

1.3. Принцип работы больших данных

Прежде всего, данные собираются на сетевых хранилищах, которые могут быть как разрозненными, так и объединены в единую систему. Информация в обязательном порядке дублируется для исключения возможных потерь и для неё характерно отсутствие структуры, т.е. это может быть текст, изображения в различных форматах, голос, музыка и т.п. В дальнейшем данные обрабатываются алгоритмом, написанным программистами для извлечения информации в удобном для человека виде. После работы алгоритма человек или группа людей делает аналитику в области обработанных данных и принимают решения о дальнейших действиях. Всё больше набирает популярность «черных ящиков» в технологии «машинного обучения» (deep learning), когда исключается человек при анализе данных и принятии решений о необходимых действиях.

1.4. Методики анализа больших данных

Наука о данных (англ. Data Science), изучающая вопросы получения, обработки, анализа информации и предоставления этих данных в понятной форме набирает популярность [4, 3]. Одним из плодов развития науки о данных получила перспективная технология «машинного обучения», так называемые чёрные ящики deep learning. Существует множество разнообразных методик анализа массивов данных, в основе которых лежит инструментарий,

заимствованный из статистики и информатики. Ниже представлен неполный список таких методик:

A/B testing. Методика, в которой контрольная выборка поочередно сравнивается с другими. Тем самым удается выявить оптимальную комбинацию показателей для достижения, например, наилучшей ответной реакции потребителей на маркетинговое предложение. Большие данные позволяют проанализировать огромное количество итераций и таким образом получить статистически достоверный результат.

Deep learning (глубинное обучение) – набор алгоритмов машинного обучения, которые пытаются моделировать высокоуровневые абстракции в данных, используя архитектуры, состоящие из множества нелинейных трансформаций.

Association rule learning. Набор методик для выявления взаимосвязей, т.е. ассоциативных правил, между переменными величинами в больших массивах данных. Используется в data mining.

Classification. Набор методик, которые позволяют предсказать поведение потребителей в определенном сегменте рынка (принятие решений о покупке, отток, объем потребления и проч.). Используется в data mining.

Cluster analysis. Статистический метод классификации объектов по группам для выявления наперед не известных общих признаков. Используется в data mining.

Crowdsourcing. Методика сбора данных из большого количества источников.

Data fusion and data integration. Набор методик, который позволяет анализировать комментарии пользователей социальных сетей и сопоставлять с результатами продаж в режиме реального времени.

Data mining. Набор методик, который позволяет определить наиболее восприимчивые для продвигаемого продукта или услуги категории потребителей, выявить особенности наиболее успешных работников, предсказать поведенческую модель потребителей.

Ensemble learning. В этом методе задействуется множество предикативных моделей за счет чего повышается качество сделанных прогнозов.

Genetic algorithms. В этой методике возможные решения представляют в виде «хромосом», которые могут комбинироваться и мутировать. Как и в процессе естественной эволюции, выживает наиболее приспособленная особь.

Machine learning. Направление в информатике (исторически за ним закрепилось название «искусственный интеллект»), которое преследует цель создания алгоритмов самообучения на основе анализа эмпирических данных.

Natural language processing (NLP). Набор заимствованных из информатики и лингвистики методик работы с естественным языком человека.

Network analysis. Набор методик анализа связей между узлами в сетях. Применительно к социальным сетям позволяет анализировать взаимосвязи между отдельными пользователями, компаниями, сообществами и т.п.

Optimization. Набор численных методов для редизайна сложных систем и процессов для улучшения одного или нескольких показателей. Помогает в принятии стратегических решений, например, состава выводимой на рынок продуктовой линейки, проведении инвестиционного анализа и проч.

Pattern recognition. Набор методик с элементами самообучения для предсказания поведенческой модели потребителей.

Predictive modeling. Набор методик, которые позволяют создать математическую модель наперед заданного вероятного сценария развития событий. Например, анализ базы данных CRM-системы на предмет возможных условий, которые подтолкнут абоненты сменить провайдера.

Regression. Набор статистических методов для выявления закономерности между изменением зависимой переменной и одной или несколькими независимыми. Часто применяется для прогнозирования и предсказаний. Используется в data mining.

Sentiment analysis. В основе методик оценки настроений потребителей лежат технологии распознавания естественного языка человека. Они позволяют

вычленив из общего информационного потока сообщения, связанные с интересующим предметом (например, потребительским продуктом). Далее оценить полярность суждения (позитивное или негативное), степень эмоциональности и проч.

Signal processing. Заимствованный из радиотехники набор методик, который преследует цель распознавания сигнала на фоне шума и его дальнейшего анализа.

Spatial analysis. Набор отчасти заимствованных из статистики методик анализа пространственных данных – топологии местности, географических координат, геометрии объектов. Источником больших данных в этом случае часто выступают геоинформационные системы (ГИС).

Statistics. Наука о сборе, организации и интерпретации данных, включая разработку опросников и проведение экспериментов. Статистические методы часто применяются для оценочных суждений о взаимосвязях между теми или иными событиями.

Supervised learning. Набор основанных на технологиях машинного обучения методик, которые позволяют выявить функциональные взаимосвязи в анализируемых частично-размеченных массивах данных.

Simulation. Моделирование поведения сложных систем часто используется для прогнозирования, предсказания и проработки различных сценариев при планировании.

Time series analysis. Набор заимствованных из статистики и цифровой обработки сигналов методов анализа повторяющихся с течением времени последовательностей данных. Одни из очевидных применений – отслеживание рынка ценных бумаг или заболеваемости пациентов.

Unsupervised learning. Набор основанных на технологиях машинного обучения методик, которые позволяют выявить скрытые функциональные взаимосвязи в анализируемых массивах данных. Имеет общие черты с Cluster Analysis.

Визуализация. Методы графического представления результатов анализа больших данных в виде диаграмм или анимированных изображений для упрощения интерпретации и облегчения понимания полученных результатов.

Наглядное представление результатов анализа больших данных имеет принципиальное значение для их интерпретации. Не секрет, что восприятие человека ограничено, и ученые продолжают вести исследования в области совершенствования современных методов представления данных в виде изображений, диаграмм или анимаций.

1.5. Аналитический инструментарий

На 2011 год некоторые из перечисленных в предыдущем подразделе подходов или определенную их совокупность позволяют реализовать на практике аналитические движки для работы с большими данными [4]. Из свободных или относительно недорогих открытых систем анализа Big Data можно порекомендовать:

1010data;

Apache Chukwa;

Apache Hadoop;

Apache Hive;

Apache Pig!;

Jaspersoft;

LexisNexis Risk Solutions HPCC Systems;

MapReduce;

Revolution Analytics (на базе языка R для мат.статистики).

JasperReports – это Java-библиотека для создания отчётов. На основе XML-шаблонов отчётов генерируются готовые для печати документы, консолидирующие данные из различных источников, в том числе JDBC. Отчёты могут выводиться на экран, принтер, либо в форматы PDF, RTF, HTML, XLS, CSV и XML. Основной частью шаблона является SQL-запрос, который указывает, какие данные необходимо выбрать из базы данных для генерации

отчёта. В XML-шаблоне допустимо применение стилей и выражений (для проверки или вычисления результата). Также возможна группировка данных по требуемому полю.

HPCC (High Performance Computing Cluster) – суперкомпьютерная платформа с открытым кодом для обработки больших данных. Компания разработчик LexisNexis Risk Solutions позиционирует ее для решения сложных задач, связанных с оценкой рисков. Высокая производительность позволяет свести время вычисления результата до нескольких секунд.

MapReduce – программный фреймворк, представленный компанией Google, используемый для параллельных вычислений над очень большими, несколько петабайт, наборами данных в компьютерных кластерах.

Язык R ориентирован на решение задач статистики и аналитики, а дистрибутив Revolution R отличается от стандартных версий с открытым исходным кодом оптимизацией для работы с большими объемами данных в многопроцессорной системе. Revolution Analytics участвует в разработке ядра языка R и некоторых других проектов на его основе, в том числе ParallelR и RНадоор. Поддержка языка R уже реализована в сервисах машинного обучения системы Azure и среде разработки MLStudio. В компании обещают и дальше участвовать в разработке R. Microsoft применяет R и в собственной работе. В частности, на нем написаны программы, подбирающие пользователям игрового сервиса Xbox Live партнеров, подходящих по уровню игры и другим факторам.

Система агрегации и анализа больших данных Apache Chukwa построена на основе Hadoop Distributed File System (HDFS) и платформы MapReduce. Таким образом, Chukwa наследует свойственные Hadoop масштабируемость и надежность. Включает мощный гибкий инструментарий для отображения результатов анализа.

Хранилище данных для Apache Hadoop. Облегчает обобщение информации, нерегламентированные запросы и анализ больших объемов данных, размещенных в совместимых с Apache Hadoop файловых системах.

Реализованный в Hive механизм позволяет осуществлять запросы на SQL-подобном языке HiveQL.

Платформа Apache Pig! для анализа больших объемов данных. Основной характерной особенностью, как и большинства аналогов, является хорошая распараллеливаемость вычислений.

Особый интерес в этом списке представляет Apache Hadoop – ПО с открытым кодом, которое за последние пять лет испытано в качестве анализатора данных большинством трекеров акций. Как только Yahoo открыла код Hadoop сообществу с открытым кодом, в ИТ-индустрии незамедлительно появилось целое направление по созданию продуктов на базе Hadoop. В настоящее время практически все современные средства анализа больших данных предоставляют средства интеграции с Hadoop. Их разработчиками выступают как стартапы, так и общеизвестные мировые компании.

1.6. Ведущие поставщики

Интерес к инструментам сбора, обработки, управления и анализа больших данных проявляли едва ли не все ведущие ИТ-компании, что вполне закономерно. Во-первых, они непосредственно сталкиваются с этим феноменом в собственном бизнесе, во-вторых, большие данные открывают отличные возможности для освоения новых ниш рынка и привлечения новых заказчиков [4].

Amazon – Интернет ресурс по продаже товаров и услуг;

Odnoklassniki – социальная сеть;

Dell – производитель ПК и серверного оборудования;

Yandex – поисковая система с предоставлением различных Интернет-сервисов;

eBay – Интернет ресурс по продаже товаров;

EMC – крупная компания по предоставлению продуктов, услуг и решений для хранения и управления информацией;

ВКонтакте – социальная сеть;

Facebook – социальная сеть;

Fujitsu – крупная японская корпорация, производитель электроники и ИТ-компания;

Google – поисковая система с предоставлением различных Интернет-сервисов;

Hitachi Data Systems Corporation предоставляет решения для хранения и виртуализации данных;

HP (Hewlett-Packard) – компания, работающая в области ИТ, предоставляющая услуги и оборудование (печать, ПК, серверы и пр.);

IBM – крупный производитель ИТ-техники;

LinkedIn – социальная сеть для деловых связей;

Microsoft – производитель программного обеспечения, мировой лидер.

NetApp – производитель сетевых хранилищ;

Oracle – лидер в производстве систем управления базами данных (СУБД);

SAP – производитель программного обеспечения для организаций с целью автоматизации деятельности;

SAS – разработчик технологического программного обеспечения и приложений;

SGI (Silicon Graphics Inc) – компания по производству графических терминалов;

Teradata – корпорация, специализирующаяся на разработке и поставке аппаратно-программных комплексов для обработки и анализа данных;

VMware – компания, крупнейший разработчик программного обеспечения для виртуализации рабочих столов, сетевых хранилищ, серверов и пр.;

Yahoo – поисковая система с предоставлением различных Интернет-сервисов.

1.7. Примеры использования больших данных

1.7.1. Case 1: Магазин «Target», США, 2012 год

Магазин «Таргет» с помощью карточек лояльности клиентов запустил в своей базе больших данных алгоритм, отслеживающий изменения поведения людей при совершении покупок в различных ситуациях [2]. Система могла самостоятельно принимать решения по рассылке рекламы того или иного продукта каждому клиенту в зависимости от изменения поведения этого клиента.

Весной 2012 года в США разгорелся скандал, связанный с получением одной семьёй рекламных буклетов с товарами для беременных женщин. Возмущённый отец дочери, которая была ученицей старших классов одной из школ, привлёк прессу. Через две недели результаты теста показали, что девушка действительно была беременна, но на момент получения рекламных буклетов не обладала этой информацией.

Таким образом, «самообучаемая машина» успела отреагировать раньше на изменение поведения девушки при покупке товаров, вызванных биологическими изменениями и отнесла её к категории беременных женщин.

1.7.2. Case 2: Браслеты сна от «Jawbone»

Производитель браслетов, отслеживающих фазы сна, продал несколько сотен тысяч единиц товара по всему миру. В каждую единицу времени активно примерно 100'000 браслетов, передающих информацию о сне своего обезличенного хозяина в базу данных компании «Джубон». Данный прибор, благодаря регистрации микродвижений руки человека фиксирует различные фазы сна, его нарушения и отклонения. По мнению руководителя организации, настоящий бизнес заключается не в продаже браслетов, а в продаже больших данных. Данная информация оказалась очень хорошим товаром для рекламных агентств, медицинских учреждений, государственных органов [2].

1.7.3. Case 3: Ancestry.com – построение семейной истории человечества

На данном ресурсе было собрано огромное количество информации по трём основным потокам [2, 3]:

- Записи из церковно-приходских книг
- 5 млн. анкет профилей пользователей
- Анализ ДНК некоторого количества пользователей
- Анализ ДНК умерших исторических личностей

Несмотря на неотвратимые ошибки при заполнении анкет пользователями, алгоритм построения взаимных связей позволил определить родственные связи всех пользователей с родственниками «вглубь» до нескольких веков.

В результате построения такой семейной истории человечества, с различной долей вероятности была получена следующая информация:

- Генеалогические деревья и установление родства.
- Миграция глобальная и внутри страны.
- Историческая картина демографии.

Стоит отметить, что данные не являются полными и в основном относятся к США и странам Европы.

1.7.4. Сайт знакомств eHarmony

Сайт eHarmony (matchaffinity.com) имеет 40 млн. зарегистрированных пользователей. При заполнении анкеты, пользователю даётся возможность заполнить около 1000 атрибутов. Повышение процента заполнения анкеты позволяет повысить вероятность автоматического подбора подходящей пары. В день данная система делает 100 млн. предположений. Благодаря обратной связи пользователей, о длительности отношений, количестве и качестве свиданий, регистрации отношений (женитьба) формируется метрика успеха. Данная метрика используется для «машинного обучения» и дальнейшее качество работы больших данных растёт. Кроме предоставления услуг, данный ресурс выявил некоторые интересные закономерности [2].

На Рис. 1 наблюдаем статистику создания пар в зависимости от разницы в росте. Вероятность успеха увеличивается, если мужчина выше женщины на 10-20 см.

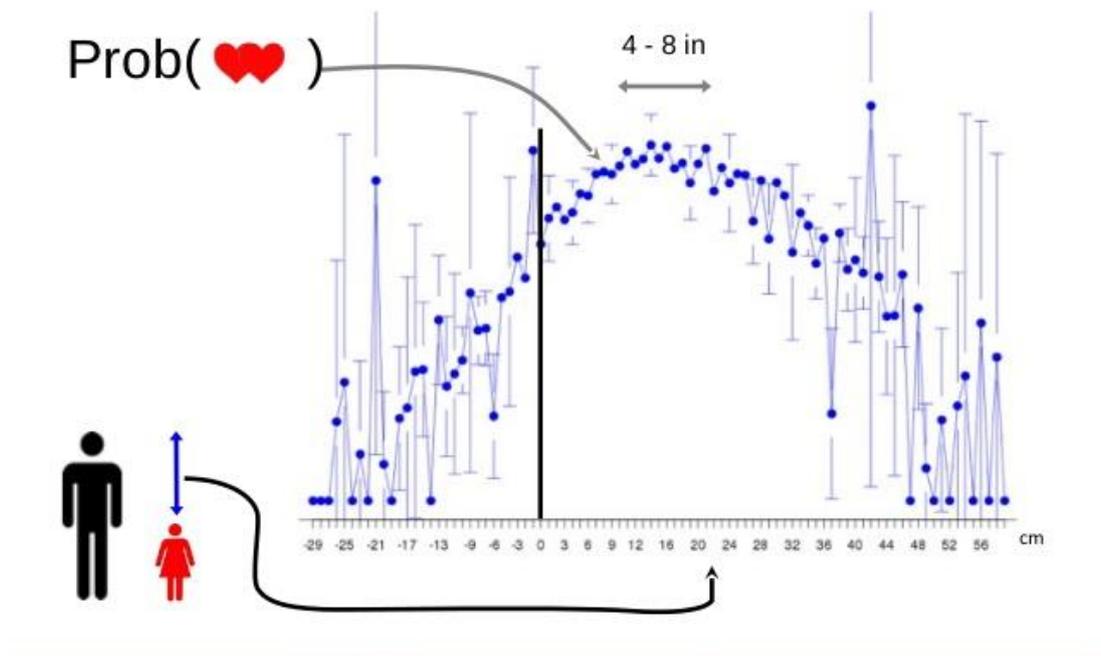


Рис. 1. Вероятность успеха создания пары в зависимости от разницы в росте партнёров

Данная статистика не противоречит бытовому представлению о том, как должна выглядеть пара. Полученные данные о взаимосвязи готовности партнёров встречаться и расстояния между их местом проживания дают нам интересную нелинейную зависимость Рис. 2. Вероятность образования пары снижается до показателя расстояния в приблизительно 100 км и держится приблизительно на одном уровне до показателя в 6500 км, а потом немного растёт.

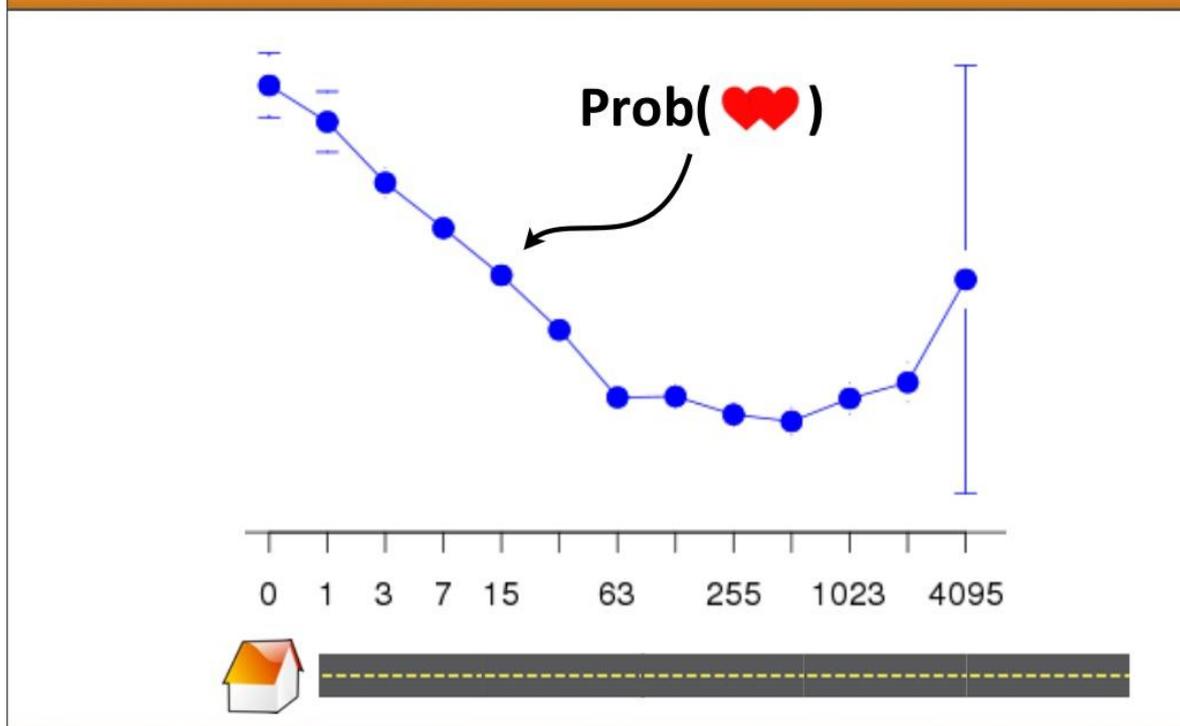


Рис. 2. Взаимосвязь готовности партнёров встречаться и расстояния между их местом проживания в милях

Ещё одна интересная зависимость вероятности долгих взаимоотношений и предпочтений в пище партнёров отображена на Рис. 3. Из этой статистики видно, что самая высокая вероятность успеха, если оба человека являются вегетарианцами и самая низкая вероятность, если оба партнёра употребляют в основном пищу из сетей быстрого питания.

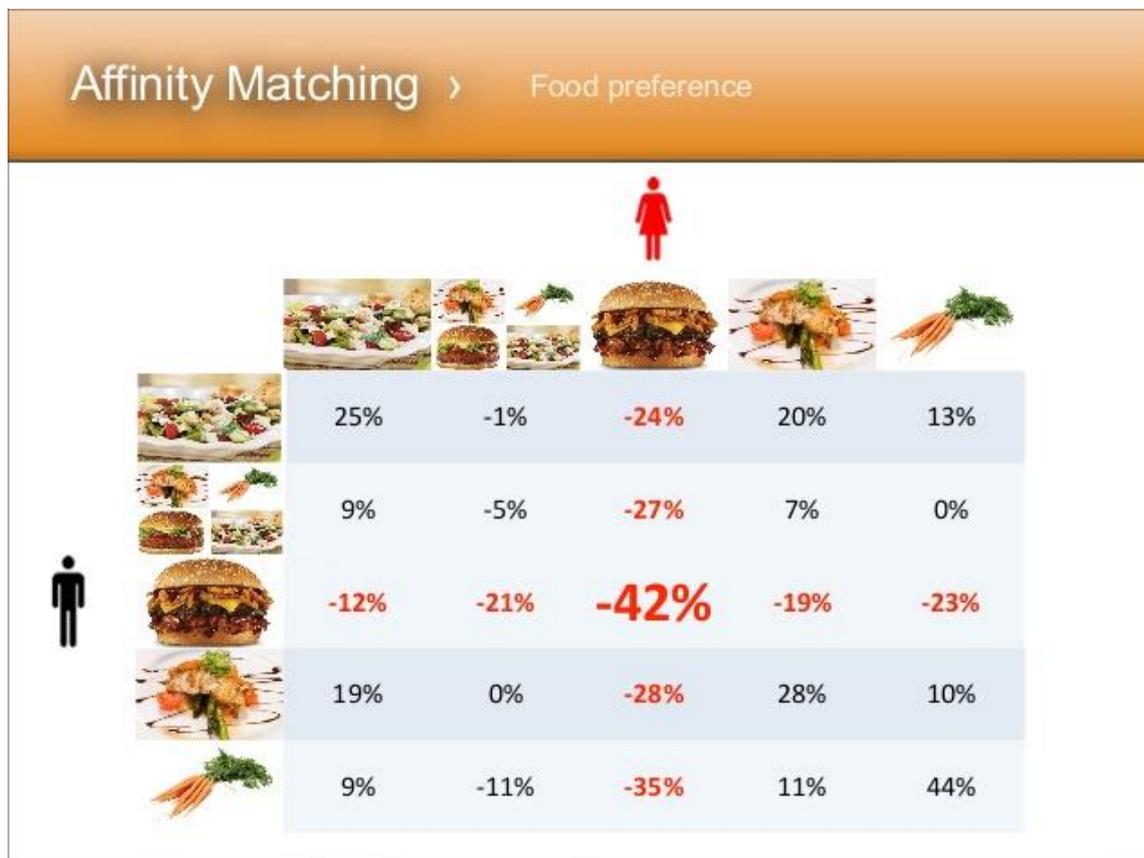


Рис. 3. Вероятность образования удачных пар и их предпочтений в пище

1.7.5. Case 4: Манипуляция настроением пользователей Facebook

На ресурсе Facebook был проведён эксперимент по манипуляции настроением пользователей. Посетители сайта были разделены на две фокус группы. Одной группе были рекомендованы к прочтению позитивные новости и позитивные комментарии других пользователей, другой группе – негативный контент. Было обнаружено, что под действием информации с определённым вектором направленности менялся эмоциональный окрас комментариев в каждой группе в соответствии с получаемой информацией [2].

1.8. Понятия Deep Learning и нейронная сеть

Глубинное или глубокое обучение (англ. Deep learning) — набор алгоритмов машинного обучения, основанный на нейросетях, которые пытаются моделировать высокоуровневые абстракции в данных, используя архитектуры, состоящие из множества нелинейных преобразований [5].

Глубокое обучение является частью более широкого семейства методов машинного обучения, которые подбирают представление данных. Наблюдение (например, изображение) может быть представлено многими способами, такими как вектор интенсивности значений на пиксель, или (в более абстрактной форме) как множество примитивов, областей определенной формы, и т. д. Некоторые представления позволяют легче решать поставленные задачи (например, распознавание лица или распознавание выражения лица). Применение глубокого обучения автоматизирует сам процесс выбора и настройки признаков, проводя обучение признаков без учителя или с частичным привлечением учителя, используя для этого эффективные алгоритмы и иерархическое извлечение признаков.

Искусственная нейронная сеть (ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Также есть мнение, что ИНС и нейронная сеть живых организмов имеют гораздо меньше общего, чем об этом принято считать. Тем не менее, наблюдается увеличение сферы применения искусственных сетей.

Метод глубокого машинного обучения получает всё большее распространение в сравнении с другими методами машинного обучения. Такая ситуация связана с тем, что в «традиционном» машинном обучении человеку необходимо самостоятельно выбрать из всего объёма информации только необходимые данные для решения конкретной задачи. Такие данные называются признаками. Если такие признаки выявить не удаётся, то машинное обучение работает плохо. Нейронные сети автоматически отделяют нужные

данные от ненужных и умеют определять правильные признаки. К минусам можно отнести потребность в больших вычислительных мощностях в связи с необходимостью обрабатывать большой объём данных и необходимость огромного набора данных для обучения, сбор которого и составляет в настоящее время наиболее трудоёмкую и требовательную к квалификации часть работы.

Необходимо отметить, что для успешной работы нейронных сетей является необходимостью их обучения на большом объеме данных.

1.9. Искусственные нейроны и их объединение в сеть

Биологический нейрон имеет ядро, в котором накапливается электрический заряд [6, 7]. Заряд поступает через короткие отростки нейрона дендриты от других нейронов. После накопления определённого заряда в ядре нейрон выдаёт электрический разряд другим нейронам через длинный отросток аксон. Этот аксон в свою очередь прикрепляется к дендритам других нейронов с помощью синапсов, которые влияют на изменение сигнала. Если сигнал при прохождении через синапс увеличивается, то такой синапс называется возбуждающим, если сигнал ослабевает – тормозящим. На Рис. 4 изображён нейрон головного мозга.

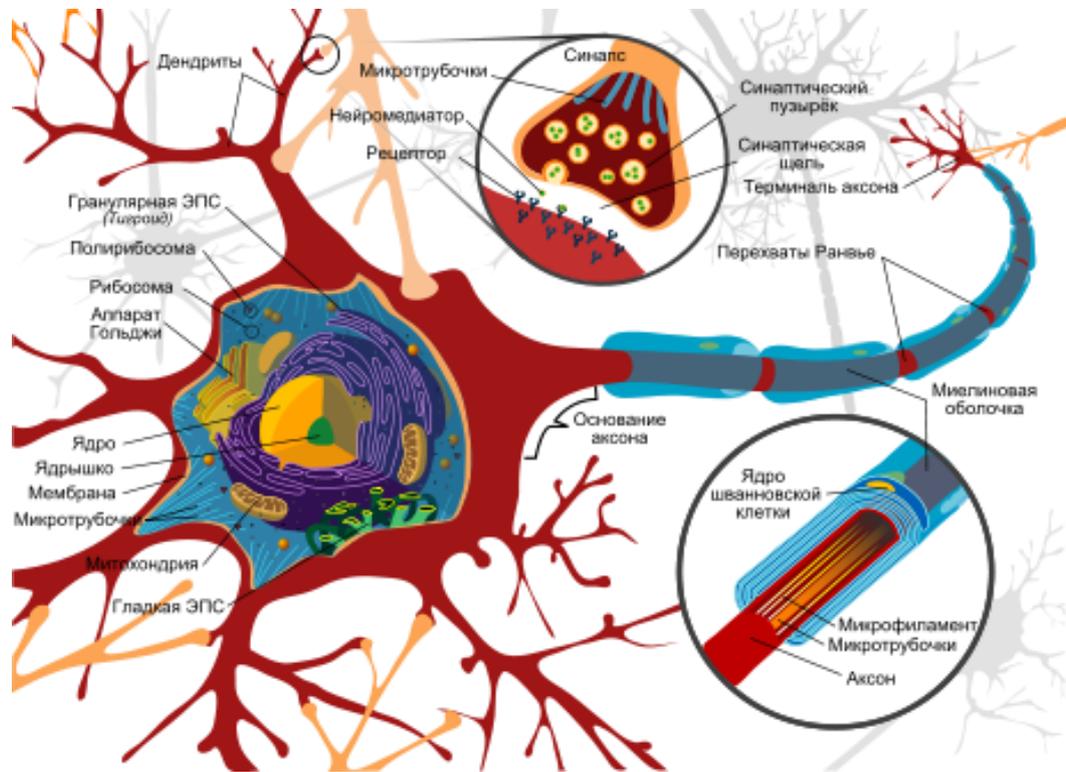


Рис. 4. Биологический нейрон

По аналогии с биологическим нейроном в 1943 году была предложена модель искусственного нейрона Мак-Каллока–Питтса Рис. 5.

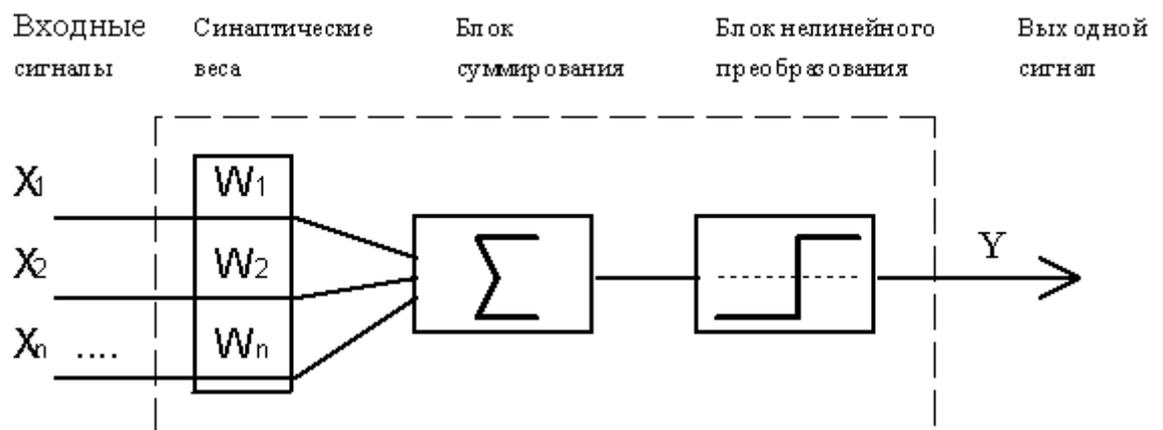


Рис. 5. Схема искусственного нейрона

На входы искусственного нейрона поступают сигналы, а каждые входы имеют свои веса, как и синапсы по аналогии с дендритами биологического нейрона. Большой положительный вес работает как возбуждающий синапс, а отрицательный вес играет роль тормозящего синапса. Далее эти сигналы поступают в блок суммирования, где сигналы складываются с учётом весов. Далее сигнал поступает в блок управления активацией, где происходит вычисление по нелинейной функции и определяется – будет ли срабатывать

нейрон на выдачу выходного сигнала, т.е. будет ли срабатывать активация нейрона и какой «мощности» сигнал будет выдаваться следующему нейрону в НС.

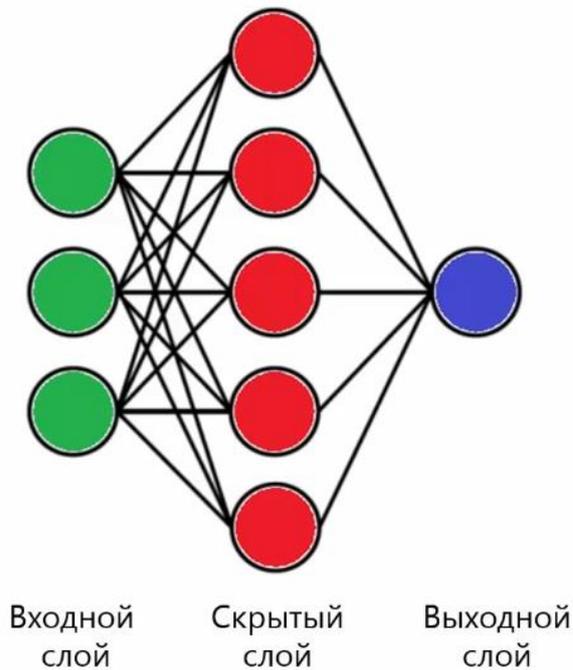


Рис. 6. Схема нейронной сети

Мак-Каллок и Питтс также предложили модель нейронной сети, изображенной на Рис. 6. Входной слой нейронной сети получает сигналы из внешнего мира, выходной слой отдаёт сигналы во внешний мир, скрытый слой обрабатывает информацию и называется скрытым, т.к. он недоступен напрямую из внешнего мира.

1.10. Архитектура нейронных сетей

На данный момент не существует единых стандартов по классификации нейронных сетей и систем глубокого обучения. Нейронные сети можно выделить в следующие архитектуры (структуры):

- Однослойные сети прямого распространения
- Многослойные сети прямого распространения
- Рекуррентные сети

Также нейронные сети выделяют по способу обучения, который тесно связан с архитектурой. Примеры способов обучения будут рассмотрены в следующем подразделе.

1.10.1. Однослойные сети прямого распространения

В многослойной НС нейроны располагаются по слоям. Однослойные сети представляют собой простейший случай, в котором существует входной и выходной слой. Именно выходной слой нейронов является единственным для вычислений. При подсчёте числа слоёв, входной слой не принимается во внимание именно из-за отсутствия на этом слое вычислений. На Рис. 7 показан пример однослойной сети. Математически нейрон однослойная сеть формирует линейное преобразование входного сигнала, которое потом обрабатывается триггером (нелинейной функцией).

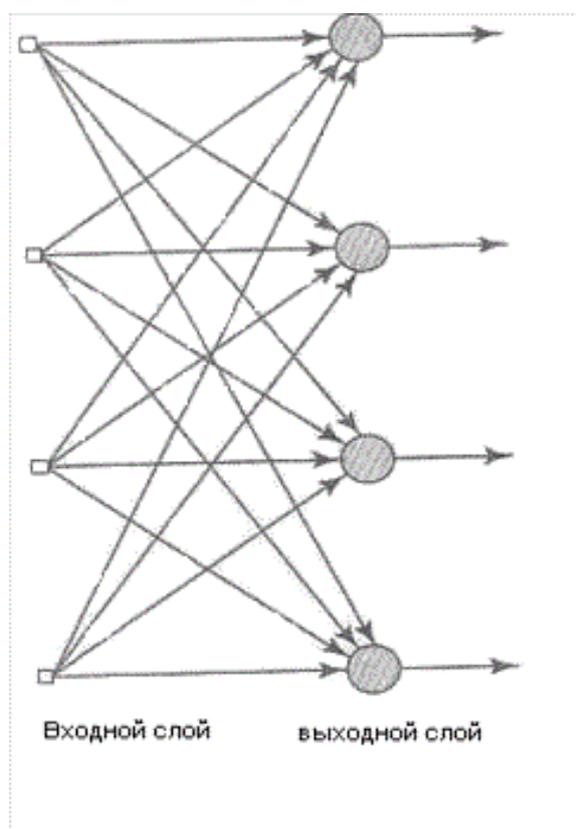


Рис. 7. Однослойная сеть прямого распространения

1.10.2. Многослойные сети прямого распространения

Данный класс нейронных сетей характеризуется наличием одного и более скрытых слоёв, узлы которых называются скрытыми. Функции последних заключаются в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько слоёв, можно выделить статистики высокого порядка. Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счёт наличия

дополнительных синаптических связей и повышая уровень взаимодействия нейронов. Способность скрытых нейронов выделять статистические зависимости высокого порядка особенно существенна, когда размер входного слоя достаточно велик.

Узлы источника входного слоя сети формируют соответствующие элементы шаблона активации (входной вектор), которые составляют входной сигнал, поступающий на нейроны (вычислительные элементы) второго слоя (т.е. первого скрытого). Выходные сигналы второго слоя используются как входные сигналы для третьего слоя и т.д. Обычно нейроны каждого из слоёв сети используют в качестве входных сигналов выходные сигналы нейронов только предыдущего слоя. Набор выходных сигналов нейронов выходного (последнего) слоя сети определяет общий отклик сети на данный входной образ, сформированный узлами источника первого входного слоя. На Рис. 8 изображена полносвязная нейронная сеть, т.е. все узлы (нейроны) следующего слоя связаны со всеми узлами предыдущего слоя. Если некоторые из синаптических связей отсутствуют, принято считать, что такая сеть называется неполносвязной.

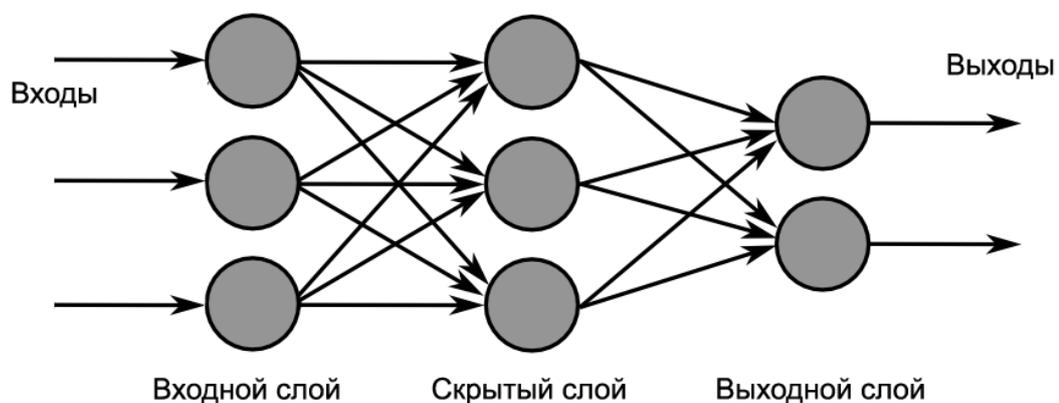


Рис. 8. Многослойная сеть прямого распространения

1.10.3. Рекуррентные сети

Рекуррентная нейронная сеть отличается от сети прямого распространения наличием, по крайней мере, одной обратной связи. Например, рекуррентная сеть может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы всех остальных

нейронов слоя, а также на собственный вход. Иными словами к рекуррентным НС сетям относятся такие сети, в которых разрешены циклы, что позволяет сети помнить предыдущее состояние.

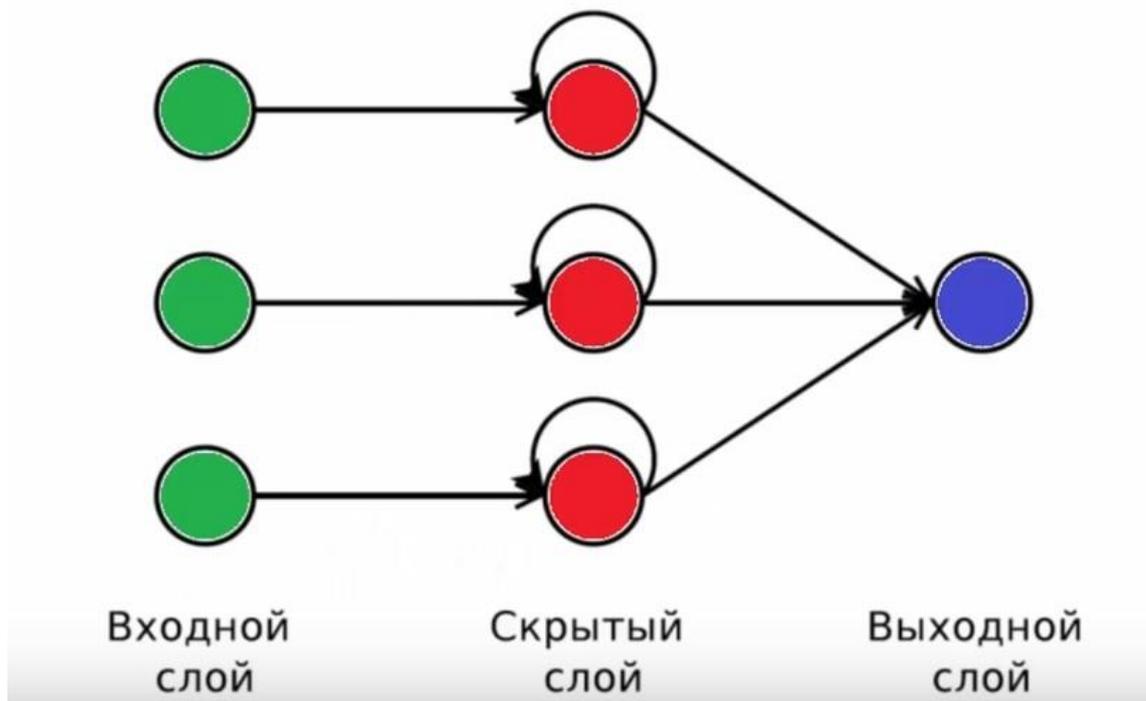


Рис. 9. Рекуррентная нейронная сеть

11.1. Способы обучения нейронных сетей

Обучение нейронной сети – это процесс, в котором параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки параметров. Различают алгоритмы обучения с учителем и без учителя.

Процесс обучения с учителем представляет собой предъявление сети выборки обучающих примеров. Каждый образец подается на входы сети, затем проходит обработку внутри структуры НС, вычисляется выходной сигнал сети, который сравнивается с соответствующим значением целевого вектора, полученного от учителя, представляющего собой требуемый выход сети.

Для того, чтобы нейронная сети была способна выполнить поставленную задачу, ее необходимо обучить. Различают алгоритмы обучения с учителем и без учителя.

Процесс обучения с учителем представляет собой предъявление сети выборки обучающих примеров. Каждый образец подается на входы сети, затем проходит обработку внутри структуры НС, вычисляется выходной сигнал сети, который сравнивается с соответствующим значением целевого вектора, представляющего собой требуемый выход сети. Затем по определенному правилу вычисляется ошибка, и происходит изменение весовых коэффициентов связей внутри сети в зависимости от выбранного алгоритма. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

Обучение без учителя (англ. Unsupervised learning, самообучение, спонтанное обучение) – один из способов машинного обучения, при котором испытуемая система спонтанно обучается выполнять поставленную задачу, без вмешательства со стороны экспериментатора. С точки зрения кибернетики, является одним из видов кибернетического эксперимента. Как правило, это пригодно только для задач, в которых известны описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами.

Обучение без учителя часто противопоставляется обучению с учителем, когда для каждого обучающего объекта принудительно задаётся «правильный ответ», и требуется найти зависимость между стимулами и реакциями системы. Например, когда нейронной сети предоставляют массив фотографий людей и машин, то по структуре объектов сеть без учителя сама распознаёт их как разные категории при этом не относя их к конкретным категориям типа «люди» и «машины».

Также можно выделить ещё один способ – обучение с подкреплением. Это способ, когда нейронная сеть действует как агент во внешней среде, не имея чётких ответов о правильности выводов, получая только сигнал о правильности выполнения действий.

Существует много типов искусственных нейронных сетей по способу обучения. Далее кратко рассмотрим некоторые из них.

11.1.1. Спайковые сети

Это особый класс сетей, в которых сигнал представлен не вещественным числом, а набором импульсов (спайков) одинаковой амплитуды и длительности, и информация содержится не в амплитуде, а в интервалах между импульсами, в их паттерне. Спайковые нейроны на выходе генерируют спайки, либо одиночные (если суммарный вход не очень большой), или пакеты (если суммарный вход достаточно велик). Этот тип сетей почти полностью копирует процессы, проходящие в мозгу человека, единственное серьезное отличие — для обучения не придумано ничего лучше правила Хебба (которое звучит примерно так: если второй нейрон сработал сразу после первого, то связь от первого ко второму усиливается, а если сразу перед первым — то ослабевает), для которого был придуман ряд небольших усовершенствований, но, к сожалению, повторить свойства мозга в области обучения пока толком не получилось.

Сети такого типа умеют приспособливаться для решения различных задач, решаемых другими сетями, но редко результаты оказываются существенно лучше. В большинстве случаев удается только повторить уже достигнутое.

Сильные стороны: очень интересны для изучения как модели биологических сетей.

Слабые стороны: почти любое практическое применение выглядит необосновано, сети других типов справляются не хуже.

Перспективы: моделирование масштабных спайковых сетей в ближайшие годы вероятно даст много ценной информации о психических расстройствах, позволит классифицировать нормальный и ненормальный режимы работы различных отделов мозга. В более отдаленной перспективе, после создания подходящего алгоритма обучения, такие сети по функциональности сравняются или даже превзойдут другие типы нейросетей, а еще позднее на их основе

можно будет собирать структуры, пригодные для прямого подключения к биологическому мозгу, для расширения возможностей интеллекта.

11.1.2. Метод Розенблатта

В случае однослойной сети желаемые выходные значения нейронов единственного слоя известны, и подстройка весов синаптических связей идет в направлении, минимизирующем ошибку на выходе сети.

По этому принципу построен алгоритм Розенблатта. Данный метод был предложен Ф.Розенблаттом в 1959 г. для искусственной нейронной сети, названной перцептрон. Перцептрон имеет пороговую функцию активации, его схема представлена здесь.

Весовые коэффициенты меняются только в том случае, если реальное выходное значение не совпадает идеальным выходным значением.

11.1.3. Многослойный перцептрон

Самая известная и очень старая архитектура, в которой идут подряд несколько слоев нейронов – входной, один или несколько скрытых слоев, и выходной слой. Почти всегда обучается методом обратного распространения ошибки – что автоматически означает, что мы должны предоставить для обучения набор пар «входной вектор – правильный выход». Тогда входной вектор отправится на вход сети, последовательно будут рассчитаны состояния всех промежуточных нейронов, и на выходе образуется выходной вектор, который сравнивается с правильным. Расхождение даёт ошибку, которую можно распространить обратно по связям сети, вычислить вклад в итоговую ошибку каждого нейрона, и скорректировать его веса, чтобы ее исправить. Повторив эту процедуру много тысяч раз, возможно, выйдет обучить сеть.

Сеть такого типа обычно очень хорошо справляется с задачами, где:

- ответ действительно зависит только от того, что мы даем на вход сети, и никак не зависит от истории входов (т.е. это не динамический процесс, или, по крайней мере, мы дали на вход исчерпывающую информацию об этом процессе в форме, пригодной для обработки сетью);

- ответ не зависит/слабо зависит от высоких степеней и/или произведений параметров – функции этого типа сеть строить почти не умеет;
- в наличии есть достаточно много примеров (желательно иметь не менее сотни примеров на каждую связь сети), или у вас есть большой опыт борьбы с эффектом специализации или «переобучением». Это связано с тем, что имея много коэффициентов, сеть может банально запомнить много конкретных примеров, и выдавать на них отличный результат – но ее прогнозы не будут иметь ничего общего с реальностью в случае, если дать на вход примеры не из обучающей выборки.

Сильные стороны – изучена со всех сторон, хорошо работает на своих задачах, если на некоторой задаче не работает (действительно не работает, а не по криворукости, как это бывает чаще всего) – то это повод утверждать, что задача сложнее, чем казалось.

Слабые стороны – неумение работать с динамическими процессами, необходимость большой обучающей выборки.

Перспективы – никаких существенных. Большинство серьезных задач, которые все еще требуют решения, не входят в класс задач, решаемых многослойным перцептроном с методом обратного распространения ошибки.

11.1.4. Рекуррентный перцептрон

На первый взгляд похож на обычный перцептрон, единственное существенное отличие состоит в том, что его выходы попадают ему же на входы, и участвуют в обработке уже следующего входного вектора. То есть, в случае рекуррентного перцептрона имеет место не набор отдельных, ничем не связанных образов, а некоторый процесс, и значение имеют не только сами входы, но и то, в какой последовательности они поступают. Из-за этого возникают отличия в методе обучения – используется то же самое обратное распространение ошибки, но для того, чтобы ошибка попала по рекуррентной связи в прошлое, используются разные ухищрения (если подойти к задаче «в лоб» – то возникнет проблема ухода ошибки на бесконечное число циклов назад). В остальном же ситуация похожа на обычный перцептрон – для

обучения нужно иметь достаточно длинную последовательность пар вход-выход, которую нужно много раз прогнать через сеть, чтобы ее обучить (или же иметь под рукой мат. модель искомого процесса, которую можно гонять во всевозможных условиях, и в реальном времени давать результаты сети для обучения).

Сеть такого типа обычно хорошо решает задачи управления динамическими процессами (начиная от классической задачи стабилизации перевернутого маятника, и до любых систем, которыми вообще хоть как-то получается управлять), предсказания динамических процессов, кроме курса валют, и вообще всего, где помимо явно наблюдаемого входа у системы есть некоторое внутреннее состояние, которое не совсем понятно как использовать.

Сильные стороны: сеть очень хороша для работы с динамическими процессами.

Слабые стороны: если все же не работает, понять, в чем проблема – очень затруднительно, в процессе обучения может вылететь в самовозбуждение (когда сигнал, полученный с выхода, забивает все, что приходит по входам), если решение все же получено – сложно понять, можно ли добиться лучших результатов, и каким путем. Другими словами, плохо изучена.

Перспективы: этот подход явно не исчерпал себя в вопросах управления – собственно, на данный момент рекуррентные перцептроны используются довольно редко, хотя их потенциал высок. Интересные результаты может дать подход с непрерывно адаптирующейся к объекту управления сетью, хотя для этого еще необходимо решить проблему неустойчивости обучения.

11.1.5. Ассоциативная память

Это широкий класс сетей, которые в той или иной степени напоминают архитектуру Хопфилда, которая состоит из одного слоя нейронов, выходы которого поступают на его входы в следующий момент времени. Этот слой служит и входом сети (в начальный момент выходы нейронов принимаются равными входному вектору), и ее выходом – значения на нейронах, образовавшиеся в конце работы, считаются ответом сети. Эта сеть меняет свои

состояния с течением времени до тех пор, пока состояние не перестанет меняться. Свойства весовой матрицы выбраны таким образом, чтобы устойчивое состояние всегда гарантированно достигалось (и обычно это происходит за несколько шагов). Такая сеть помнит некоторое количество векторов, и при подаче на вход любого вектора, может определить, на какой из запомненных он более всего похож – отсюда и название. Двухслойная модификация этой сети (гетероассоциативная память) может запоминать вектора не по-одному, а по парам разной размерности.

Сети такого типа хорошо справляются с задачами, где нужно определить похожесть вектора на один из стандартных запомненных. Собственно, это единственный класс задач, где они хороши. Также конкретно сеть Хопфилда может использоваться для решения задач оптимизации (например, задачи комивояжера), однако ее эффективность в этой области под вопросом.

Сильные стороны – очень быстрое обучение (т.к. вместо градиентного спуска решается система уравнений), возможность удалить образ из памяти или добавить в память, не затронув остальные, некоторые свойства такой памяти напоминают свойства мозга, и их изучение интересно с такой позиции.

Слабые стороны – очень узкий класс решаемых задач, неумение обобщать примеры, максимальный объем памяти жестко связан с размерностью запоминаемого вектора (ввиду особенностей построения).

11.1.6. Нейронные сети Кохонена

Нейронные сети Кохонена типичный пример нейросетевой архитектуры, обучающейся без учителя. Отсюда и перечень решаемых ими задач: кластеризация данных или прогнозирование свойств. Кроме того, сети Кохонена могут использоваться с целью уменьшения размерности данных с минимальной потерей информации.

Нейронные сети, которые обучались с учителем на выборках данных, включающих множество примеров, состоящих из соответствующих друг другу пар входных и выходных векторов. При этом выходные значения принимали самое непосредственное участие в настройке весовых коэффициентов. В

нейронных сетях Кохонена выходные вектора в обучающей выборке могут быть, но могут и отсутствовать, и, в любом случае, они не принимают участия в процессе обучения. То есть выходы не используются в качестве ориентиров при коррекции синапсов. Именно поэтому данный принцип настройки нейронной сети называется самообучением.

В рассматриваемой архитектуре сигнал распространяется от входов к выходам в прямом направлении. Структура нейронной сети содержит единственный слой нейронов (слой Кохонена) без коэффициентов смещения. Нейронные сети Кохонена типичный пример нейросетевой архитектуры, обучающейся без учителя. Отсюда и перечень решаемых ими задач: кластеризация данных или прогнозирование свойств. Кроме того, сети Кохонена могут использоваться с целью уменьшения размерности данных с минимальной потерей информации.

Количество нейронов равно количеству кластеров, среди которых происходит начальное распределение и последующее перераспределение обучающих примеров. Количество входных переменных нейронной сети равно числу признаков, характеризующих объект исследования и на основе которых происходит отнесение его к одному из кластеров.

Следует различать собственно самообучение и самоорганизацию нейронной сети Кохонена. При обычном самообучении сеть имеет строго фиксированную структуру, т. е. количество нейронов, не изменяющееся на протяжении всего жизненного цикла. При самоорганизации сеть, напротив, не имеет постоянной структуры. В зависимости от найденного расстояния до нейрона-победителя либо этот нейрон используется для кластеризации примера, либо для поданного на входы примера создается новый кластер с соответствующими ему весовыми коэффициентами. Кроме того, в процессе самоорганизации структуры сети Кохонена отдельные нейроны могут исключаться из нее.

11.1.7. Свёрточные сети

Свёрточные нейронные сети (СНС) – одни из самых влиятельных инноваций в области компьютерного зрения. Впервые нейронные сети привлекли всеобщее внимание в 2012 году, когда Алекс Крижевски благодаря им выиграл конкурс ImageNet (иными словами, это ежегодная олимпиада по машинному зрению), снизив рекорд ошибок классификации с 26% до 15%, что тогда стало прорывом. Сегодня глубинное обучение лежит в основе услуг многих компаний: Facebook использует нейронные сети для алгоритмов автоматического проставления тегов, Google – для поиска среди фотографий пользователя, Amazon – для генерации рекомендаций товаров, Pinterest – для персонализации домашней страницы пользователя, а Instagram – для поисковой инфраструктуры.

Но классический, и, возможно, самый популярный вариант использования сетей это обработка изображений.

11.2. Сферы использования глубинного обучения в нейронных сетях

Не смотря на то, что технология глубокого обучения существует десятки лет, широкое распространение эта технология получила только в последние годы. В первую очередь это вызвано двумя тенденциями: доступность больших данных в цифровом виде и возможностью параллельных вычислений на центральном и графическом процессорах (изобретение компании NVIDIA в 2007 году). Сейчас глубинное обучение применяют в таких различных областях, как промышленность, образование, наука, медицина, реклама и т.п. Также нейронные сети применяют для различных развлекательных целей, например в приложениях «Prisma» и «Artisto» (Рис. 10), когда по фотографиям или видео генерируется изображение в стиле известных художников.

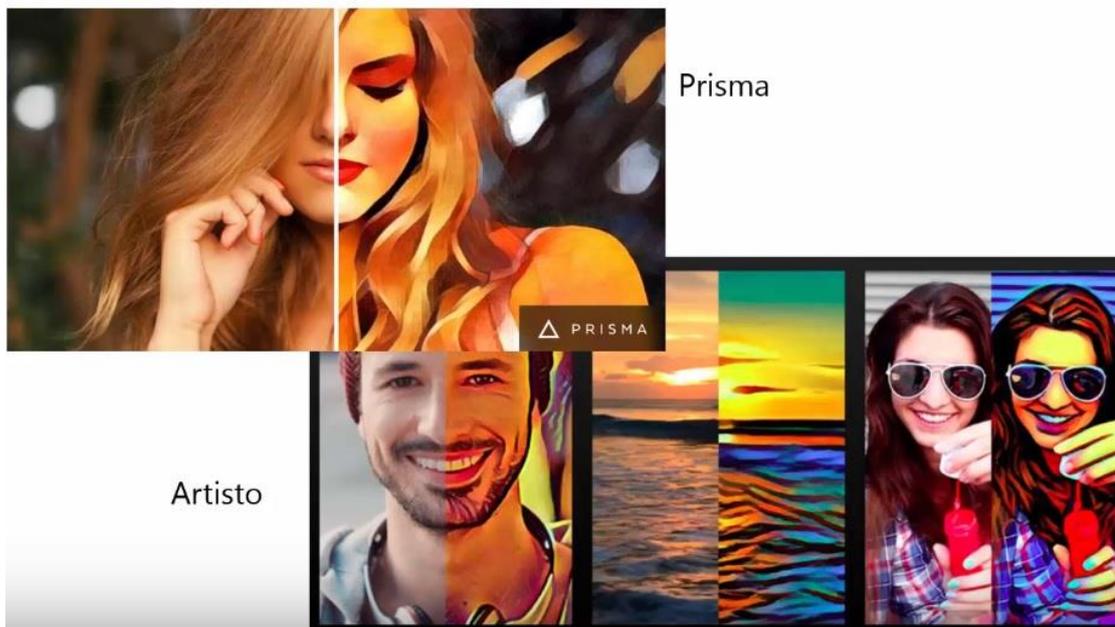


Рис. 10. Применение нейронных сетей для развлекательных целей

Также у нейронных сетей есть большое количество практических применений:

- перевод текстов с иностранных языков;
- синхронный перевод голоса собеседников, общающихся на разных языках, во время разговора через приложение;
- управление автомобилем без водителей;
- электронные роботы консьержи в общественных местах для общения с клиентами, применяя естественное общение и распознавание эмоций;
- в медицинских целях (диагностика повреждений, постановка диагноза, анализ эффективности лечения, очки для слепых и т.п.);
- в экономике используют для оценки рисков, прогнозе стоимости валют, распознавание финансовых документов, безопасность банковских операций и пр.;
- авионика: обучаемые автопилоты, распознавание сигналов радаров, адаптивное пилотирование сильно поврежденного самолета, беспилотные летательные аппараты;
- безопасность и охранные системы: распознавание лиц; идентификация личности по отпечаткам пальцев, голосу, подписи или лицу; распознавание автомобильных номеров, анализ аэрокосмических снимков, мониторинг

информационных потоков в компьютерной сети и обнаружение вторжений, обнаружение подделок, анализ данных с видеодатчиков и разнообразных сенсоров.

11.3. Перспективы по модернизации нейронных сетей

В данное время ведутся постоянные работы по совершенствованию нейронных сетей, которые всё больше проникают во многие сферы человеческой жизни. Несомненно, что в будущем, благодаря искусственным нейронным сетям будет увеличиваться степень автоматизации, что неизбежно приведёт к исчезновению многих профессий человека. На мой взгляд, бояться этого не нужно, т.к. человечество проходило через это и ничего страшного в целом, если не считать отдельных человеческих трагедий, не произошло. Тем не менее, мы можем столкнуться с лавинным отказом от использования «живого» труда, что может вызвать экономические проблемы и социальную напряжённость в обществе. Чтобы перестроиться в новых условиях, уже сейчас необходимо моделировать общество будущего и место человека в нём.

Научное и околонучное сообщество осознаёт, что существующие искусственные нейронные сети очень далеки от биологических нейронных сетей даже по архитектуре. Новые исследования в области изучения мозга дают представление о векторе развития ИНС.

Одна из моделей принципа работы нейронной сети мозга, которая мне показалась очень перспективной для рассмотрения возможности применения в искусственных сетях, была представлена Алексеем Редозубовым [8, 9, 10, 11].

Согласно этой модели работа мозга ближе к принципу работы компьютера нежели к искусственной нейронной сети. Активность нейронов формирует код, с которым в дальнейшем работает всё нейронное пространство. Нет привязки активности конкретного нейрона при реакции на конкретную информацию, т.к. один и тот же нейрон может реагировать как на похожую информацию, так и на совершенно разную. Реакция биологических нейронов сравнивается с реакцией элементов компьютера, которые формируют код. Рассматривается понятие клеточных автоматов формирующие уникальные

узоры из волны активных нейронов по всей сети и передача информации такими узорами. При этом каждый нейрон имеет свой «список» комбинаций соседних активных нейронов для «понимания» в каких случаях он сам должен активироваться.

Предполагается, что мозг работает не с аналоговыми, а с дискретными сигналами, таким образом, мозг (кора мозга) работает с конечным числом дискретных понятий. Вводится понятие интерференции информационной волны, способствующей формированию ключа воспоминаний (время, пространство, обстоятельства и пр.) для работы памяти с информацией, которую необходимо запомнить. Т.е. по аналогии с компьютером, когда для хранения информации используются различные метки (путь к файлу, время создания, различные теги и т.п.), для того, чтобы потом можно было легко найти файл даже по части имеющейся в ключе информации.

При этом сохранение информации происходит не в конкретном месте, а распределено по всему мозгу. Причём подразумевается распределение не частей информации по разным местам, а сохранение всего жизненного объёма информации во множестве мест в головном мозге в кортикальных столбах.

Одним из фактов, в пользу данной концепции являются эксперименты Карла Лешли в течение тридцати лет над обученными выходить из лабиринта крысами, которым последовательно удаляли части мозга в поиске места хранения информации с данным навыком. Эксперименты показали, что не существует отдельных частей мозга, где хранились бы конкретные воспоминания, вся память дублирована в различных частях мозга. Повторением экспериментов другими учёными также не удалось обнаружить, что целостная память находится в определённом месте мозга [12].

Допускаю, что ИНС могут эволюционировать нейронные сети нового типа, согласно концепции А. Редозуба, что с большой долей вероятности приблизит человечество к созданию искусственного интеллекта.

11.4. Применение нейронных сетей в сфере обучения

Как уже отмечалось ранее, успешное применение нейронных сетей можно реализовать посредством её обучения большим количеством данных. Важно понимать, что применение технологии Deep Learning отдельными педагогами практически невозможно при отсутствии технической поддержки со стороны специалистов информационных технологий (ИТ), а также без достаточного финансирования. Тем не менее, как показывает практика, внедрение данной технологии даёт серьёзный экономический эффект и повышение качества продукции (товаров, услуг).

11.4.1. Обработка результатов тестирования учащихся

Одним из вариантов использования технологии машинного обучения является «техническое зрение», когда результаты тестирования учащихся обрабатывает нейронная сеть. Наибольшего эффекта можно добиться при обработке результатов Единого Государственного Экзамена (ЕГЭ). Благодаря тому, что существует огромный материал за предыдущие года, можно создать хорошую библиотеку для обучения и приблизить точность распознавания ответов в новых работах к 100%, а время обработки сократить в разы.

Используя эту же нейронную сеть, существуют перспективы повысить процент успешной сдачи ЕГЭ, за счёт внедрения бесплатного сервиса тренировочных экзаменов для учащихся за прошлые периоды. Данный сервис будет работать круглосуточно и позволит оперативно получать результаты и позволит максимально приблизить экзаменуемых к реальным условиям, а также даст больше свободы действий преподавателям на местах в различных регионах.

11.4.2. Выявление эффективных методов обучения

Одним из важных факторов эффективного обучения является готовность и возможность учащихся воспринимать информацию. В силу устоявшихся моделей поведения и, в силу возраста, низкого эмоционального интеллекта учащихся, зачастую невозможно выявить готовность последних к восприятию информации. Нейронная сеть, анализируя мимику и невербальные знаки, может

успешно справляться с выявлением у обучаемых людей отсутствие концентрации в связи с усталостью, потерей интереса, нарушением логической последовательности получаемой информации и т.п. Данная информация может быть полезна как в конкретных случаях для преподавателей и служить триггером, так и для формирования массива данных в технологии Big Data для последующего анализа эффективности различных методик и методов обучения посредством корреляции с успеваемостью учащихся.

Стоит отметить, что применение Deep Learning и Big Data с высокой долей вероятности позволит выявить корреляцию между методом обучения и различными характеристиками обучаемых групп, например, возраст обучаемых, предметная область, уровень заданий, пол, социальное положение и т.п.

Раздел 12: ПРАКТИЧЕСКАЯ РАБОТА ПО ОБУЧЕНИЮ НЕЙРОННОЙ СЕТИ

12.1. Задача и необходимые инструменты для практического выполнения глубинного обучения нейронной сети 1

Задача: Обучить свёрточную нейронную сеть распознавать графические изображения цифр в различных вариантах исполнения с точностью выше 95% [13, 14, 15].

Для решения практической задачи по глубинному обучению нейронной сети были применены следующие инструменты:

- язык программирования «Python 3.5»
- дистрибутив Anaconda, включающий в себя интерпретатор языка «Python 3.5» и различные математические пакеты
- библиотеки «Keras» и «Theano»
- персональный компьютер с операционной системой «Windows 7»
- подключение к сети Интернет для получения доступа к необходимому

ПО

12.1.1. Установка ПО «Anaconda Python»

Для установки программного обеспечения, в котором будет производиться практическая работа по обучению нейронной сети необходимо загрузить дистрибутив «Anaconda», который находится по ссылке <https://www.continuum.io/downloads> и запустить установку:

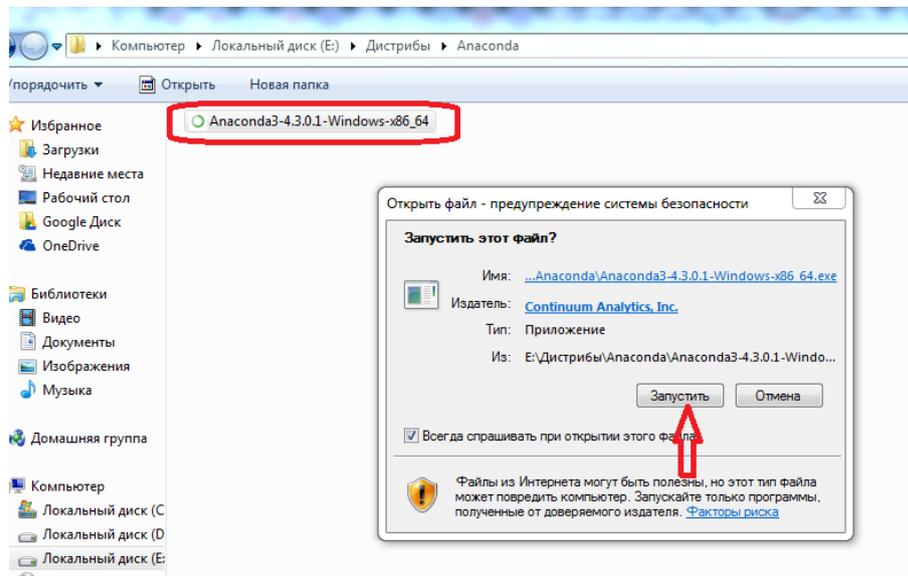


Рис. 11. Запуск дистрибутива «Anaconda»

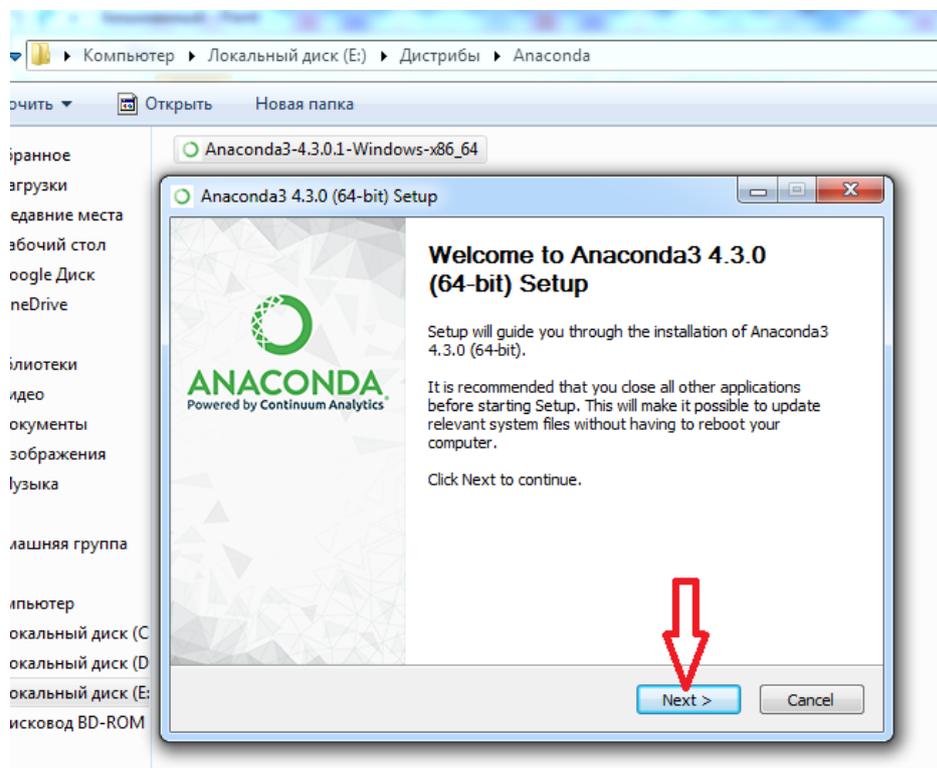


Рис. 12. Начало установки дистрибутива «Anaconda»

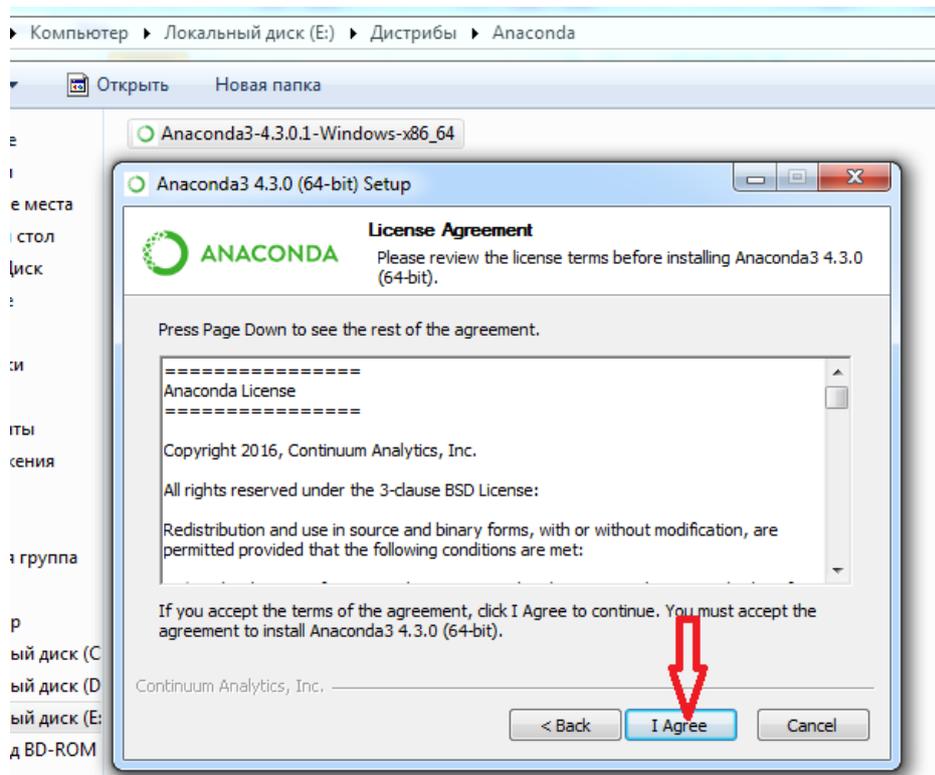


Рис. 13. Согласие с условиями использования ПО «Anaconda»

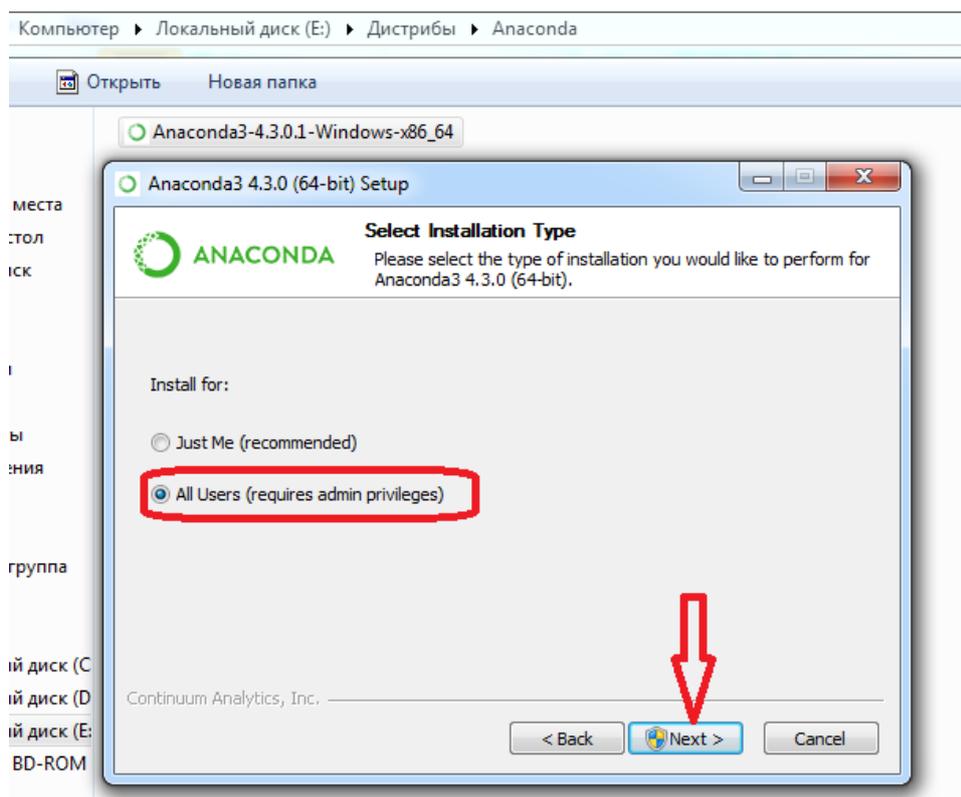


Рис. 14. Предоставление доступа всем пользователям к ПО «Anaconda»

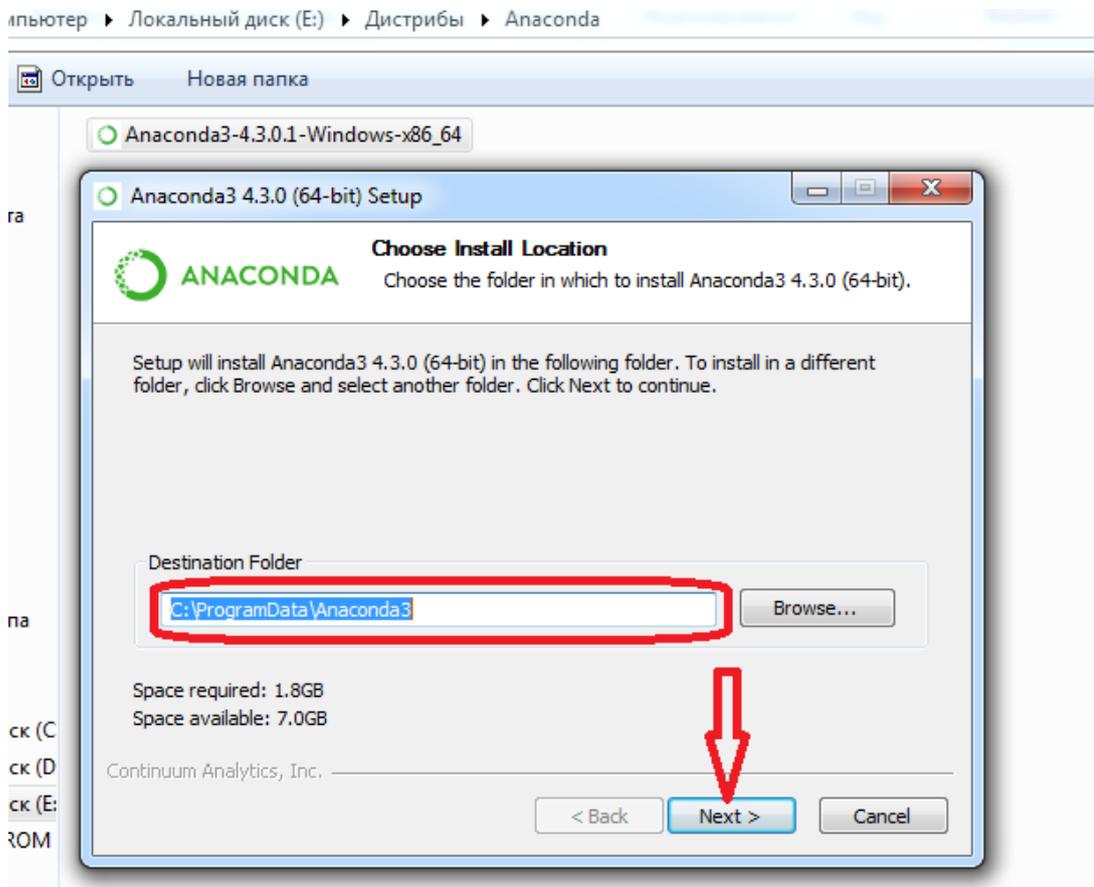


Рис. 15. Запуск процесса установки ПО «Anaconda»

После отобразится прогресс установки «Anaconda» и по его завершению необходимое программное обеспечение будет установлено на компьютере.

12.1.2. Установка пакетов «Theano» и «Keras» в «Anaconda»

Через «Пуск», «Все программы» можно увидеть инсталлированное ПО Anaconda. Для установки необходимых пакетов «Theano» и «Keras» необходимо запустить консоль «Anaconda Prompt».

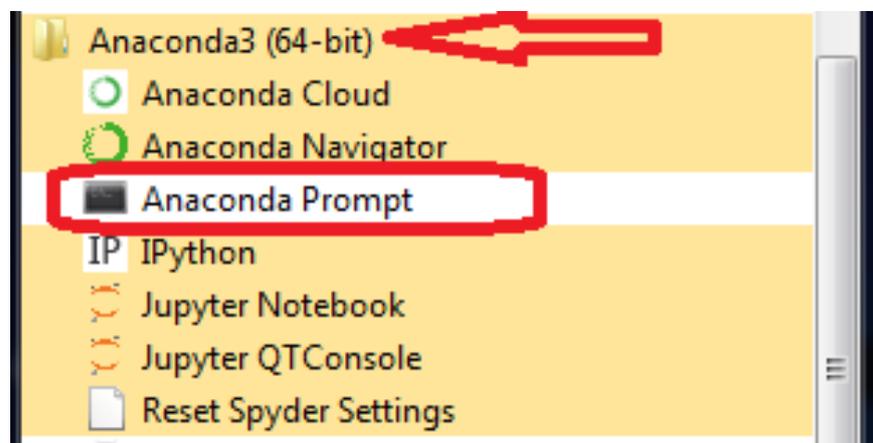


Рис. 16. Запуск консоли «Anaconda»

Для установки пакета «Theano» прописываем в консоли команду **conda install theano** и нажимаем «Ввод». Также можно использовать команды копирования и вставки:

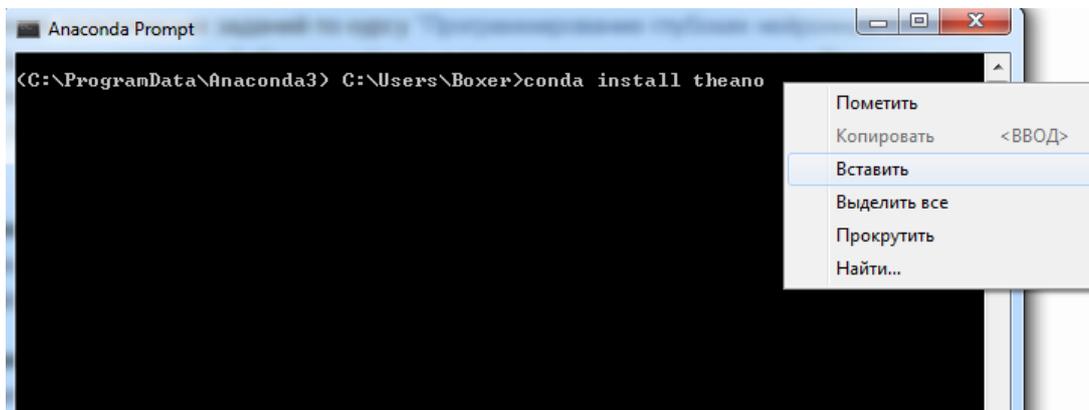


Рис. 17. Подключение пакета «Theano»

Для установки пакета «Keras», который не входит в основной набор пакетов «Anaconda» прописываем в консоли команду **conda install -c conda-forge keras** и нажимаем «Ввод».

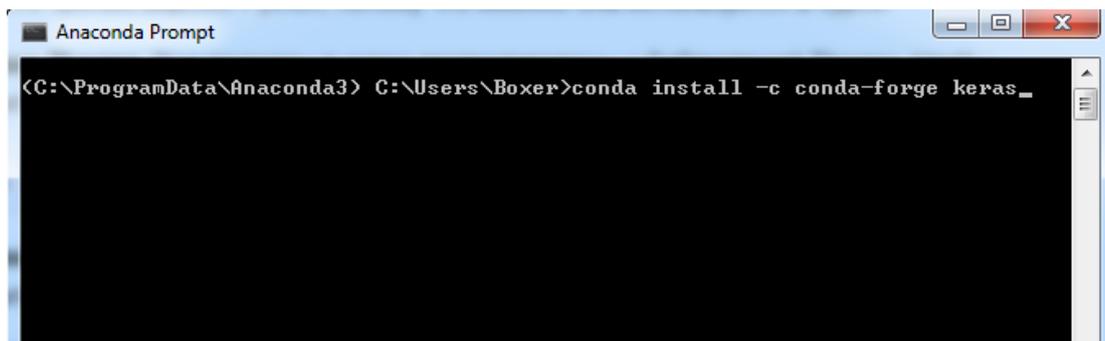


Рис. 18. Подключение пакета «Keras»

В файле `.keras/keras.json`, который находится в домашнем каталоге пользователя, прописываем «Theano» в качестве бекенда:

```
{  
    "epsilon": 1e-07,  
    "backend": "theano",  
    "image_dim_ordering": "th",  
    "floatx": "float32"  
}
```

Рис. 19. Параметры пакета «Keras»

Также указываем, что будем использовать порядок хранения измерений в изображениях, который применяется в Theano ("image_dim_ordering": "th"). В TensorFlow используется другой порядок.

12.1.3. Установка набора данных «MNIST»

Для обучения нейронной сети потребуется набор данных «MNIST», который содержит два файла, в одном из которых 60 тысяч изображений рукописных цифр, в другом – метки классов, т.е. правильные ответы к ним. Загрузить данный пакет можно, перейдя по адресу <http://yann.lecun.com/exdb/mnist>.



Рис. 20. Пример массива изображений из пакета «MNIST»

Изображения записаны в бинарном виде, формат изображений 28x28 пикселей, оттенки серого от 0 (белый цвет) до 255 (чёрный цвет).

12.1.4. Обучение нейронной сети

Нейронная сеть для MNIST имеет два слоя. Входной слой имеет 800 нейронов, на который подаётся 784 значения (28x28 пикселей). Выходной слой имеет 10 нейронов и вероятность того, что на изображении определённая цифра.

Для начала работ по непосредственному обучению нейронной сети запускаем интерпретатор «Spyder» из папки «Anaconda» в меню «Пуск»:

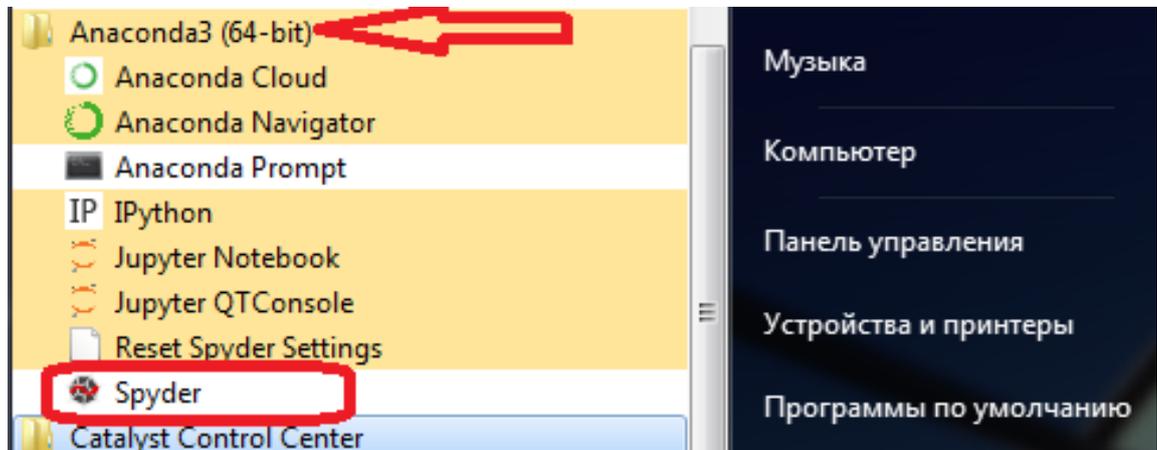


Рис. 21. Запуск интерпретатора «Spyder»

В консоли вводим программу обучения нейронной сети, как показано на Рис. 22.

```

1 #-*- coding: utf-8 -*-
2 """
3 Created on Mon Feb 27 16:39:40 2017
4
5 @author: Boxer
6 """
7 # Подключаем модуль эффективной работы с массивами в Python
8 import numpy
9 # Импорт datasets
10 from keras.datasets import mnist
11 # Подключаем модель нейронной сети Sequential, слои которой соединены друг с другом
12 from keras.models import Sequential
13 # Тип слоев нейронной сети Dense, где все нейроны предыдущего уровня соединены со всеми нейронами следующего уровня
14 from keras.layers import Dense
15 # Утилиты для работы с массивами
16 from keras.utils import np_utils
17
18 # Устанавливаем seed для повторяемости результатов
19 numpy.random.seed(42)
20
21 # Загружаем данные с алгоритмами для машинного обучения и для тестов из библиотеки Keras
22 (X_train, y_train), (X_test, y_test) = mnist.load_data()
23 # X_train - набор данных с изображениями, y_train - набор данных с правильными ответами
24
25 # Преобразование размерности изображений (входного массива)
26 X_train = X_train.reshape(60000, 784)
27 X_test = X_test.reshape(10000, 784)
28 # Нормализация данных
29 X_train = X_train.astype('float32')
30 X_test = X_test.astype('float32')
31 X_train /= 255
32 X_test /= 255
33
34 # Преобразуем метки в категории
35 Y_train = np_utils.to_categorical(y_train, 10)
36 Y_test = np_utils.to_categorical(y_test, 10)
37
38 # Создаем последовательную модель
39 model = Sequential()
40
41 # Добавляем уровни сети (два слоя)
42 model.add(Dense(800, input_dim=784, init="normal", activation="relu"))
43 model.add(Dense(10, init="normal", activation="softmax"))
44
45 # Компилируем модель с методом стохастического обучения SGD, с мерой ошибки categorical_crossentropy,
46 # с метрикой оптимизации adam
47 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=["accuracy"])
48
49 print(model.summary())
50
51 # Обучаем сеть методом fit, с уровнем градиента в 200 изображений. Обучение проходит в 100 эпох
52 model.fit(X_train, Y_train, batch_size=200, nb_epoch=100, validation_split=0.2, verbose=1)
53
54 # Оцениваем качество обучения сети на тестовых данных
55 scores = model.evaluate(X_test, Y_test, verbose=0)
56 print("Точность работы на тестовых данных: %.2f%%" % (scores[1]*100))

```

Рис. 22. Фотография экрана «Spyder» с программой обучения НС

```
# Подключаем модуль эффективной работы с массивами в Python
import numpy

# Импорт datasets
from keras.datasets import mnist

# Подключаем модель нейронной сети Sequential, слои которой
соединены друг с другом
from keras.models import Sequential

# Тип слоев нейронной сети Dense, где все нейроны предыдущего уровня
соединены со всеми нейронами следующего уровня
from keras.layers import Dense

# Утилиты для работы с массивами
from keras.utils import np_utils

# Устанавливаем seed для повторяемости результатов
numpy.random.seed(42)

# Загружаем данные с алгоритмами для машинного обучения и для тестов
из библиотеки Keras
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# X_train - набор данных с изображениями, y_train - набор данных с
правильными ответами

# Преобразование размерности изображений (входного массива)
X_train = X_train.reshape(60000, 784)
X_test = X_test.reshape(10000, 784)

# Нормализация данных
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255

# Преобразуем метки в категории
Y_train = np_utils.to_categorical(y_train, 10)
```

```

Y_test = np_utils.to_categorical(y_test, 10)
# Создаем последовательную модель
model = Sequential()
# Добавляем уровни сети (два слоя)
model.add(Dense(800, input_dim=784, init="normal", activation="relu"))
model.add(Dense(10, init="normal", activation="softmax"))
# Компилируем модель с методом стохастического обучения SGD, с
мерой ошибки categorical_crossentropy,
# с метрикой оптимизации accuracy
model.compile(loss="categorical_crossentropy", optimizer="SGD",
metrics=["accuracy"])
print(model.summary())
# Обучаем сеть методом fit, с уровнем градиента в 200 изображений.
Обучение проходит в 100 эпох
model.fit(X_train, Y_train, batch_size=200, nb_epoch=100,
validation_split=0.2, verbose=1)
# Оцениваем качество обучения сети на тестовых данных
scores = model.evaluate(X_test, Y_test, verbose=0)
print("Точность работы на тестовых данных: %.2f%%" % (scores[1]*100))

```

Пример вывода данных в процессе обучения можно увидеть на Рис. 23.

```
Epoch 1/100
60000/60000 [=====] - 3s - loss:
1.3748 - acc: 0.6791
Epoch 2/100
60000/60000 [=====] - 3s - loss:
0.7159 - acc: 0.8446
Epoch 3/100
60000/60000 [=====] - 3s - loss:
0.5465 - acc: 0.8710
...
Epoch 99/100
60000/60000 [=====] - 3s - loss:
0.1185 - acc: 0.9686
Epoch 100/100
60000/60000 [=====] - 3s - loss:
0.1177 - acc: 0.9691
```

Рис. 23. Вывод данных в процессе обучения

12.2. Задача и необходимые инструменты для практического выполнения глубинного обучения нейронной сети 2

Задача: Обучить свёрточную нейронную сеть распознавать направление взгляда учащегося. Предлагается использовать технологию, опирающуюся только на встроенные в браузер технологии и не требующую установки расширений [16].

Для решения практической задачи по глубинному обучению нейронной сети были применены следующие инструменты:

- языки разметки и программирования «HTML», «JavaScript»
- браузер «Google Chrome»
- персональный компьютер с операционной системой «Windows 7»
- видеочамера
- подключение к сети Интернет

12.2.1. Создание исходного HTML-файла

Предлагается самостоятельно собрать данные и обучить модель прямо в браузере, на собственном компьютере. При таком подходе серверная обработка данных совершенно не требуется.

Идея заключается в наблюдении за глазами пользователя с помощью видеочамеры в процессе обучения нейронной сети. Важным моментом является то, что предлагается анализировать не всю информацию, поступающую с видеочамеры, а лишь изображение лица человека, посредством «захвата» лица.

Для обнаружения лица человека используется библиотека `clmtrackr`, которая имеет маленький размер и отличается высокой производительностью. В качестве входного изображения простой нейронной свёрточной сети используется маленькое изображение, что позволяет без особых трудностей обучить эту сеть.

Для начала загрузим `clmtrackr.js` из соответствующего репозитория. Работу начнём с пустого HTML-файла, в котором импортируются `jQuery`,

TensorFlow.js, clmtrackr.js и файл main.js с нашим кодом, над которым мы будем работать немного позже:

```
<!doctype html>
<html>
<body>
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.12.0"></script>
  <script src="clmtrackr.js"></script>
  <script src="main.js"></script>
</body>
</html>
```

Рис. 24. Подготовка HTML и подключение файлов .js

12.2.2. Получение потока с видеокamеры

Для активации камеры и вывода потока видео на страницу браузера необходимо получить разрешение пользователя. Добавляем код, в ранее созданный HTML-файл, в пределах тега <body>, но выше тегов <script>:

```
<video id="webcam" width="400" height="300" autoplay></video>
```

Рис. 25. Код для активации web-камеры

Файл main.js должен содержать следующий код для вывода изображения на экран web-страницы:

```
$(document).ready(function() {
  const video = $('#webcam')[0];

  function onStreaming(stream) {
    video.srcObject = stream;
  }

  navigator.mediaDevices.getUserMedia({ video: true }).then(onStreaming);
});
```

Рис. 26. Код в файле main.js

12.2.3. Поиск лица на видеоизображении

Для поиска лица на изображении необходимо использовать библиотеку `clmtrackr.js`:

```
const ctrack = new clm.tracker();
ctrack.init();
```

Рис. 27. Инициализация системы слежения за лицом

В функции `onStreaming()`, мы подключаем систему поиска лица, добавляя туда следующую команду:

```
ctrack.start(video);
```

Рис. 28. Подключение системы поиска лица

Теперь система распознаёт лицо на видео. Для контроля распознавания лица необходимо вывести «маску». Выводим изображение маски поверх видео (элемента ответственного за показ) с помощью тега `<canvas>`, ответственного за возможность рисования в HTML:

```
<canvas id="overlay" width="400" height="300"></canvas>
<style>
  #webcam, #overlay {
    position: absolute;
    top: 0;
    left: 0;
  }
</style>
```

Рис. 29. Добавление тега `<canvas>`

Теперь, каждый раз, когда браузер выводит очередной кадр видео, необходимо рисовать что-то на элементе `<canvas>`. Выполнение какого-либо кода при выводе каждого кадра выполняется с помощью механизма `requestAnimationFrame()`. Прежде чем что-либо будет вводиться в элемент `<canvas>`, необходимо удалить с него то, что было на нём раньше, очистив его. Затем предлагаем `clmtrackr` выполнять вывод графики прямо на элемент `<canvas>`. Добавить соответствующий код надо ниже команды `ctrack.init()`:

```
<canvas id="overlay" width="400" height="300"></canvas>
<style>
  #webcam, #overlay {
    position: absolute;
    top: 0;
    left: 0;
  }
</style>
```

Рис. 30. Наложение «маски» поверх лица

После этого на изображение накладывается «маска» лица:

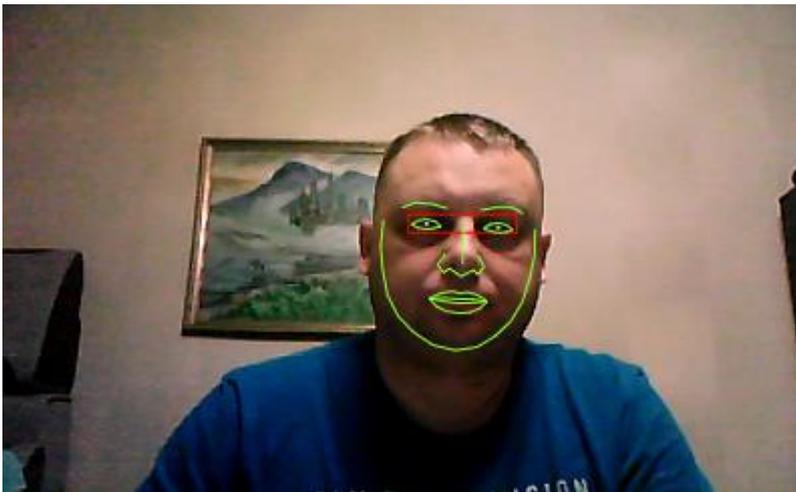


Рис. 31. «Захват» лица при обучении нейронной сети

Теперь нам нужно обнаружить прямоугольную область изображения, в которой находятся глаза, и поместить её на отдельный элемент `<canvas>`.

12.2.4. Выявление области изображения, содержащей глаза

Необходимо обнаружить прямоугольную область изображения, в которой находятся глаза, и поместить её на отдельный элемент `<canvas>`.

Cmltracker даёт не только сведения о расположении лица, но и о 70 контрольных точек. С помощью документации к cmltracker нужно выбрать именно те контрольные точки, которые нам нужны:

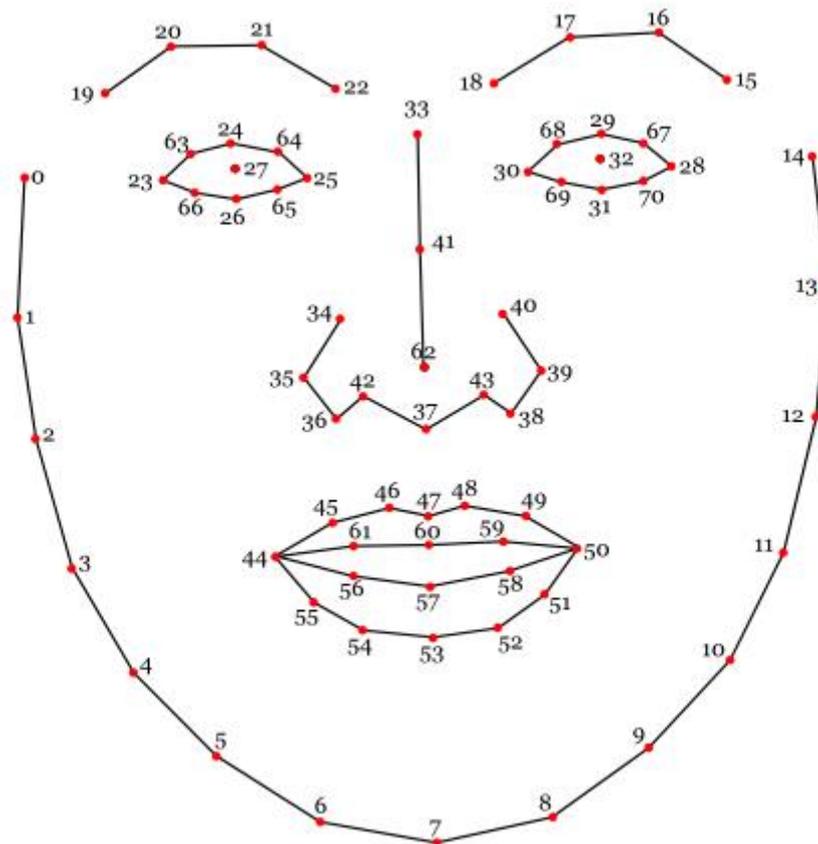


Рис. 32. Выбор контрольных точек для обнаружения глаз

Решим, что глаза — это прямоугольная часть изображения, границы которой касаются точек 23, 28, 24 и 26, расширенная на 5 пикселей в каждом направлении. Этот прямоугольник должен включать в себя всё, что для нас важно, если только пользователь не слишком сильно наклоняет голову.

Теперь, прежде чем сможем воспользоваться этим фрагментом изображения, нужен ещё один элемент `<canvas>` для его вывода. Его размеры будут равны 50x25 пикселей. Прямоугольник с глазами будет вписан в этот элемент. Небольшие деформации изображения не вызовут проблем.

Необходимо добавить в HTML-файл код, описывающий элемент `<canvas>`, в который попадёт та часть изображения, на которой имеются глаза:

```

<canvas id="eyes" width="50" height="25"></canvas>
<style>
  #eyes {
    position: absolute;
    top: 0;
    right: 0;
  }
</style>

```

Рис. 33. Код для поиска глаз на изображении лица

Следующая функция вернёт координаты x и y , а также ширину и высоту прямоугольника, окружающего глаза. Она, в качестве входных данных, принимает массив `positions`, полученный от `clmtrackr`. Каждая координата, полученная от `clmtrackr`, имеет компоненты x и y . Эту функцию надо добавить в `main.js`:

```

function getEyesRectangle(positions) {
  const minX = positions[23][0] - 5;
  const maxX = positions[28][0] + 5;
  const minY = positions[24][1] - 5;
  const maxY = positions[26][1] + 5;

  const width = maxX - minX;
  const height = maxY - minY;

  return [minX, minY, width, height];
}

```

Рис. 34. Функция по возвращению координат глаз

Теперь, в каждом кадре, мы собираемся извлекать из видеопотока прямоугольник с глазами, обводить его красной линией на элементе `<canvas>`, который наложен на элемент `<video>`, а затем копировать его в новый элемент `<canvas>`. Обратите внимание на то, что для того, чтобы правильно выявить нужную нам область, мы будем рассчитывать показатели `resizeFactorX` и `resizeFactorY`.

Замените следующим кодом блок `if` в функции `trackingLoop()`:

```

if (currentPosition) {
  ctrack.draw(overlay
  const eyesRect = getEyesRectangle(currentPosition);
  overlayCC.strokeStyle = 'red';
  overlayCC.strokeRect(eyesRect[0], eyesRect[1], eyesRect[2], eyesRect[3]);
  const resizeFactorX = video.videoWidth / video.width;
  const resizeFactorY = video.videoHeight / video.height;
  const eyesCanvas = $('#eyes')[0];
  const eyesCC = eyesCanvas.getContext('2d');
  eyesCC.drawImage(
    video,
    eyesRect[0] * resizeFactorX, eyesRect[1] * resizeFactorY,
    eyesRect[2] * resizeFactorX, eyesRect[3] * resizeFactorY,
    0, 0, eyesCanvas.width, eyesCanvas.height
  ); }

```

Рис. 35. Извлечение из видеопотока области глаз

Теперь будет видно изображение красного прямоугольника вокруг глаз, а то, что содержит этот прямоугольник – в соответствующем элементе <canvas>

12.2.5. Организация процесса сбора данных

Существует много способов сбора данных. Предлагается использовать информацию, которую можно получить от мыши и клавиатуры.

Пользователь перемещает курсор по странице и следит за ним глазами, нажимая на клавишу **Пробел** на клавиатуре каждый раз, когда программа должна записать очередной образец. При таком подходе несложно быстро собрать большой набор данных для обучения модели.

Для того чтобы узнать, где именно на веб-странице расположен указатель мыши, понадобится обработчик события `document.onmousemove`. Наша функция, кроме того, нормализует координаты таким образом, чтобы они укладывались в диапазон `[-1, 1]`:

```

// Отслеживание перемещений мыши:
const mouse = {
  x: 0,
  y: 0,

  handleMouseMove: function(event) {
    // Получим позицию указателя и нормализуем её, приведя к диапазону [-1, 1]
    mouse.x = (event.clientX / $(window).width()) * 2 - 1;
    mouse.y = (event.clientY / $(window).height()) * 2 - 1;
  },
}

document.onmousemove = mouse.handleMouseMove;

```

Рис. 36. Организация сбора данных

12.2.6. Захват изображений

Для захвата изображения, выводимого элементом `<canvas>` и сохранения его в виде тензора, TensorFlow.js предлагает вспомогательную функцию `tf.fromPixels()`. Используем её для сохранения и последующей нормализации изображения с элемента `<canvas>`, выводящего прямоугольник, содержащий глаза пользователя, как показано на Рис. 22.

```

function getImage() {
  // Захват текущего изображения в виде тензора
  return tf.tidy(function() {
    const image = tf.fromPixels($('#eyes')[0]);
    // Добавление <font color="#999999">измерения</font></i>:
    const batchedImage = image.expandDims(0);
    // Нормализация и возврат данных:
    return batchedImage.toFloat().div(tf.scalar(127)).sub(tf.scalar(1));
  });
}

```

Рис. 37. Функция для захвата изображения

Функция `tf.tidy()` используется для того, чтобы навести порядок после завершения работы.

Есть возможность сохранить все образцы в одной большой обучающей выборке, однако в машинном обучении важно проверять качество обучения модели. Именно поэтому необходимо сохранить некоторые образцы в отдельной контрольной выборке. После этого можно проверить поведение модели на новых для неё данных и узнать, не произошло ли чрезмерного обучения модели. Для этой цели 20% от общего количества образцов включены в контрольную выборку.

Код, который используется для сбора данных и формирования выборок можно увидеть на Рис. 23.

```

const dataset = {
  train: {
    n: 0,
    x: null,
    y: null,
  },
  val: {
    n: 0,
    x: null,
    y: null,
  },
}

function captureExample() {
  tf.tidy(function() {
    const image = getImage();
    const mousePos = tf.tensor1d([mouse.x, mouse.y]).expandDims(0);
    const subset = dataset[Math.random() > 0.2 ? 'train' : 'val'];
    if (subset.x == null) {
      subset.x = tf.keep(image);
      subset.y = tf.keep(mousePos);
    } else {
      const oldX = subset.x;
      const oldY = subset.y;
      subset.x = tf.keep(oldX.concat(image, 0));
      subset.y = tf.keep(oldY.concat(mousePos, 0));
    }
    subset.n += 1;
  });
}

```

Рис. 38. Вывод данных в процессе обучения

Также необходимо привязать эту функцию к клавише **Пробел**:

```

$('body').keyup(function(event) {
  // Выполняется при нажатии на клавишу Пробел на клавиатуре
  if (event.keyCode == 32) {
    captureExample();

    event.preventDefault();
    return false;
  }
});

```

Рис. 39. Вывод данных в процессе обучения

Теперь каждый раз, когда нажимают на клавишу **Пробел**, изображение глаз и координаты указателя мыши добавляются в один из наборов данных.

12.2.7. Создание нейронной сети и обучение модели

Создадим простую свёрточную нейронную сеть. TensorFlow.js предоставляет для этой цели API, напоминающее Keras. У сети должен быть слой conv2d, слой maxPooling2d и слой dense с двумя выходными значениями, которые представляют экранные координаты. Попутно необходимо добавить в сеть, в качестве регуляризатора, слой dropout, и слой flatten для того, чтобы преобразовать двухмерные данные в одномерные. Обучение сети выполняется с помощью оптимизатора Adam.

Код модели:

```

let currentModel;
function createModel() {
  const model = tf.sequential();
  model.add(tf.layers.conv2d({
    kernelSize: 5,
    filters: 20,
    strides: 1,
    activation: 'relu',
    inputShape: [$('#eyes').height(), $('#eyes').width(), 3],
  }));
  model.add(tf.layers.maxPooling2d({
    poolSize: [2, 2],
    strides: [2, 2],
  }));
  model.add(tf.layers.flatten());
  model.add(tf.layers.dropout(0.2));
  model.add(tf.layers.dense({
    units: 2,
    activation: 'tanh',
  }));
  model.compile({
    optimizer: tf.train.adam(0.0005),
    loss: 'meanSquaredError',
  });
  return model;
}

```

Рис. 40. Создание модели свёрточной нейронной сети

Прежде чем приступить к обучению сети, необходимо задать фиксированное количество эпох и переменный размер пакета, чтобы предусмотреть работу с очень маленькими наборами данных:

```

function fitModel() {
  let batchSize = Math.floor(dataset.train.n * 0.1);
  if (batchSize < 4) {
    batchSize = 4;
  } else if (batchSize > 64) {
    batchSize = 64;
  }

  if (currentModel == null) {
    currentModel = createModel();
  }

  currentModel.fit(dataset.train.x, dataset.train.y, {
    batchSize: batchSize,
    epochs: 20,
    shuffle: true,
    validationData: [dataset.val.x, dataset.val.y],
  });
}

```

Рис. 41. Код фиксированного количества эпох

Добавление на страницу кнопку для запуска обучения. Этот код идёт в HTML-файл:

```

<button id="train">Train!</button>
<style>
  #train {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 24pt;
  }
</style>

```

Рис. 42. Код кнопки запуска обучения нейронной сети (HTML)

Код для добавления в JS-файл:

```

$('#train').click(function() {
  fitModel();
});

```

Рис. 26. Код кнопки запуска обучения нейронной сети (JS-файл)

Теперь, когда есть возможность собирать данные и подготовлена модель, можно приступить к процессу предсказания места на странице, куда смотрит пользователь. Укажем на это место с помощью зелёного круга, который перемещается по экрану.

Сначала добавим на страницу круг:

```
<div id="target"></div>
<style>
  #target {
    background-color: lightgreen;
    position: absolute;
    border-radius: 50%;
    height: 40px;
    width: 40px;
    transition: all 0.1s ease;
    box-shadow: 0 0 20px 10px white;
    border: 4px solid rgba(0,0,0,0.5);
  }
</style>
```

Рис. 26. Добавление круга, перемещаемого по экрану

Для того чтобы перемещать круг по странице, периодически передаётся текущее изображение глаз в нейронную сеть и задаётся ей вопрос о том, куда смотрит пользователь. Модель в ответ выдаёт две координаты, по которым должен быть перемещён круг:

```

function moveTarget() {
  if (currentModel == null) {
    return;
  }
  tf.tidy(function() {
    const image = getImage();
    const prediction = currentModel.predict(image);

    // Конвертируем нормализованные координаты в позицию на экране
    const targetWidth = $('#target').outerWidth();
    const targetHeight = $('#target').outerHeight();
    const x = (prediction.get(0, 0) + 1) / 2 * ($(window).width() - targetWidth);
    const y = (prediction.get(0, 1) + 1) / 2 * ($(window).height() - targetHeight);

    // Переместим в нужное место кружок:
    const $target = $('#target');
    $target.css('left', x + 'px');
    $target.css('top', y + 'px');
  });
}

setInterval(moveTarget, 100);

```

Рис. 26. Обучение нейронной сети

Установлен интервал на 100 миллисекунд. В зависимости от мощности компьютера возможно изменение данного интервала.

Теперь для обучения свёрточной нейронной сети необходимо выполнять одновременно следующие действия: перемещать курсор мыши по экрану, следя за ним глазами, многократно нажимать клавишу Пробел. Далее нужно нажать кнопку запуска обучения.

Потом необходимо собирать ещё данные повторно и нажмите кнопку запуска обучения. Через некоторое время зелёный кружок начнёт передвигаться по экрану вслед за взглядом пользователя. Поначалу он будет не особенно точно попадать в то место, куда смотрит пользователь, но, начиная с примерно 50 собранных образцов, после нескольких этапов обучения, зелёный круг будет довольно точно перемещаться в ту точку страницы, на которую смотрит пользователь.

Подготовленная HTML страница с вышеописанным кодом находится по ссылке <https://cpury.github.io/lookie-lookie/>.

Раздел 13: ЭЛЕКТИВНЫЙ КУРС

13.1. Рабочая программа элективного курса БОЛЬШИЕ ДАННЫЕ (Big Data) И ГЛУБОКОЕ МАШИННОЕ ОБУЧЕНИЕ (Deep Learning) В НЕЙРОННЫХ СЕТЯХ.

Учитель: Тихонов А.А.

Автор: Тихонов А.А.

Предмет: информатика

Класс: 11

Образовательная область - **Информатика**

Количество часов: 34

13.2. Пояснительная записка

Типология курса: предметный

В современном мире информационные технологии (ИТ) играют большую роль, проникая во все сферы деятельности человека, без ИТ не представляется развитие различных областей науки, а сами ИТ обеспечивают фундаментальный компонент образования. Более того, отдых современного человека тесно связан с достижениями информационных технологий, что определяет особое положение предмета «Информатика» в системе школьного образования.

Не смотря на то, что уже с детства люди сталкиваются с информатикой, необходимо проявить максимум усилий в процессе обучения для формирования у обучаемых полного и всестороннего развития в этой области.

Вместе с другими предметами естественнонаучного и технического циклов информатика создает основу для формирования способностей к аналитическому, формально-логическому мышлению. Поиск разумного баланса между этими двумя системами целей - основной вопрос любой учебной программы и методики преподавания курса.

Большинство ВУЗов предъявляет высокие требования к абитуриентам, а потенциальные работодатели всё чаще предпочитают принимать на работу сотрудников, обладающих определённым уровнем компетенций в области информационных технологий даже на рабочих специальностях.

Обязательная школьная программа по Информатике не успевает за прогрессом в области информационных технологий, программы в высших учебных заведениях также отстают как минимум на 5 лет от потребностей обучающихся. На сегодняшний день проблема решается индивидуальным подходом отдельных преподавателей и энтузиастов, размещающих обучающие видеоролики в Интернете.

С 2012 по 2016 года количество российских вакансий с упоминанием машинного обучения (machine learning) возросло в 18 раз. Если в 2012 году таких вакансий было 113, то в 2016 – уже 2037. Количество вакансий с упоминанием big data по сравнению с 2012 годом выросло более чем в 28 раз. Если в 2012 году таких вакансий на всю Россию приходилось 83, то в 2016 году их было опубликовано 2389 [17].

Принципы отбора и организации учебного материала. Данный курс носит практическую направленность и будет требовать от учащихся самостоятельной работы на уроках при выполнении творческого задания.

Основные научные понятия: большие данные (Big Data), глубокое машинное обучение (Deep Learning), нейронной сети или искусственные нейронные сети.

Ожидаемый результат. В результате освоения курса учащийся должен: уметь работать с искусственной нейронной сетью посредством работы в среде программирования Python, знать основные методы анализа больших данных, знать виды нейронных сетей и их архитектуру, уметь анализировать эффективность и область применения нейронных сетей, понимать связь между большими данными и нейронными сетями.

Согласно рекомендациям и утверждённым стандартам данный элективный курс рассчитан на 34 часа (1 час в неделю в 11 классе), программой предусматривается последовательное изучение разделов:

1. Большие данные (Big Data)
2. Принцип работы больших данных
3. Сферы применения
4. Анализ больших данных
5. Примеры использования больших данных
6. Понятия глубокого машинного обучения (Deep Learning) и нейронной сети
7. Архитектура нейронных сетей
8. Способы обучения нейронных сетей
9. Перспективы модернизации нейронных сетей
10. Перспективы использования больших данных и нейронных сетей в образовании

Цели изучения курса

Основная цель курса – приобретение общих знаний в области Больших данных (Big Data) и погружение в раздел глубокого машинного обучения (Deep Learning) посредством практического применения нейронной сети.

Образовательные

Получение представления о больших данных и нейронных сетях.

Освоение технологии обучения нейронной сети.

Формирование умений использовать большие данные и машинное обучение в будущей профессиональной деятельности.

Развивающие

Развитие способностей аналитического мышления.

Развитие представлений о новом подходе к обработке информации.

Развитие творческих способностей в процессе моделирования нейронных сетей.

Воспитательные

Воспитание ответственности за результаты своей работы в коллективе, адекватная оценка своего вклада в общее дело.

Воспитание отношения к информационной культуре и информационной безопасности.

Воспитание информационной культуре.

Формируемые компетенции у обучаемых приведены в табл. 1.

Таблица 1

Формируемые компетенции

Компетенции	Описание
1	2
Ключевые общеобразовательные (общеучебные умения, способы деятельности; универсальные учебные действия)	Умение искать, извлекать и обрабатывать необходимую информацию из различных источников в процессе обучения нейронных сетей. <u>(Информационная компетенция).</u>
	Способность проявлять творческий и познавательный потенциал школьника при создании оригинальных решений алгоритмизации при работе с нейронными сетями и большими данными. <u>(Учебно-познавательная компетенция).</u>
	Самостоятельность и ответственность за результаты своей деятельности при создании коллективного проекта. Чувство команды, целеустремленность в процессе коллективной деятельности. Адекватная самооценка своей деятельности. <u>(Коммуникативная компетенция).</u>
	Воспитание информационной культуре. <u>(Общекультурная компетенция).</u>

1	2
<p>Предметные компетенции (предметные умения, способы деятельности)</p>	<p>Умение изображать алгоритмы в общепринятом формате.</p> <p>Представление о составных частях программного кода (циклы, условия, запись переменных, ввод вывод информации и т.п.).</p> <p><u>(Технологическая компетенция)</u></p> <p>Использование среды программирования при обучении нейронной сети – компиляция, поиск ошибок в алгоритме и программном коде, отладка.</p> <p><u>(Техническая компетенция).</u></p>
<p>Предметные компетенции (предметные умения, способы деятельности)</p>	<p>Соблюдение правил форматирования кода в программной среде (оформление) и подготовка описательной части (комментарии) для лучшего взаимодействия с другими программистами при реализации больших проектов. Соблюдение этических норм при работе с информацией и написании программного продукта – отказ от противоправной деятельности в профессиональной сфере <u>(Социокультурная компетенция).</u></p>

13.3. Тематическое планирование курса

Тематическое планирование курс приведено в табл. 2.

№ п/п	Название темы	Количество часов
1	2	3
1	Большие данные (Big Data)	3
1.1	Понятие Больших данных (Big Data). Вид, объём и структура информации	1
1.2	Поставщики больших данных	1
1.3	Тестирование на компьютере по теме	1
2	Принцип работы с большими данными	2
2.1	Сбор, обработка и хранение информации. Отличия от структурированных данных	1
2.2	Тестирование на компьютере по теме	1
3	Сферы применения	1
3.1	Сферы применения больших данных	1
4	Анализ больших данных	3
4.1	Методики анализа	1
4.2	Аналитический инструментарий	1
4.3	Тестирование на компьютере по теме	1
5	Примеры использования больших данных	2
5.1	Примеры использования больших данных в различных сферах деятельности	2
6	Понятия глубокого машинного обучения (Deep Learning) и нейронной сети	4
6.1	Искусственные нейроны и их объединение в сеть	2
6.2	Отличия от биологического нейрона	1
6.3	Тестирование на компьютере по теме	1
7	Архитектура нейронных сетей	7

1	2	3
7.1	Архитектура нейронных сетей	2
7.2	Моделирование однослойных нейронных сетей	1
7.3	Моделирование многослойных нейронных сетей	2
7.4	Тестирование на компьютере по теме	1
8	Способы обучения нейронных сетей	5
8.1	Принцип обучения нейронных сетей	2
8.2	Библиотеки для обучения нейронных сетей	1
8.3	Работа с обученной нейронной сетью. Понятие «переобучение»	1
8.4	Тестирование на компьютере по теме	1
9	Перспективы модернизации нейронных сетей	2
9.1	Теории работы биологических нейронных сетей.	1
9.2	Перспективные разработки нейронных сетей	1
10	Перспективы использования больших данных и нейронных сетей в образовании	2
10.1	Большие данные в образовании	1
10.2	Области возможного применения нейронных сетей в образовании.	1
11	Творческая работа	4
11.1	Выполнение творческого проекта по темам курса	4
12	Подведение итогов. Защита творческих проектов	2
	ВСЕГО:	34

13.4. Содержание курса

13.4.1. Большие данные (Big Data)

Понятия больших данных. Вид, объём и структура информации, скорость её обработки. История введения понятия. Источники больших данных. Тестирование на компьютере по теме.

13.4.2. Принцип работы с большими данными

Сбор, обработка и хранение информации. Отличия от структурированных данных. Причины появления нового способа работы с информацией. Тестирование на компьютере по теме.

13.4.3. Сферы применения

Сферы применения больших данных и решаемые задачи. Преимущества и недостатки перед структурированными данными. Тестирование на компьютере по теме.

13.4.5. Анализ больших данных

Методики анализа и аналитический инструментарий. Понятие «чёрного ящика». Необходимые знания и навыки для работы с большими данными. Тестирование на компьютере по теме.

13.4.6. Примеры использования больших данных

Рассмотрение кейсов эффективной работы с большими данными. Тестирование на компьютере по теме. Какие преимущества получают компании, работающие с большими данными. Перспективы технологии.

13.4.7. Понятия глубокого машинного обучения (Deep Learning) и нейронной сети

Понятия глубокого машинного обучения (Deep Learning) и нейронной сети. Виды нейронных сетей их общие и отличительные черты от биологических нейронов и биологических нейронных сетей. Тестирование на компьютере по теме.

13.4.8. Архитектура нейронных сетей

Рассмотрение архитектуры нейронных сетей. Однослойные, многослойные и рекуррентные нейронные сети. Тестирование на компьютере по теме.

13.4.9. Способы обучения нейронных сетей

Принцип обучения нейронных сетей большими данными. Веса нейронов и влияние веса на полученный результат. Примеры обучения и работы однослойной, многослойной и рекуррентной сети. Способы построения слоев и их влияние на обучение результат. Вероятности предсказаний событий обученными нейронными сетями. Тестирование на компьютере по теме

13.4.10. Перспективы модернизации нейронных сетей

Перспективы создания нового типа нейронных сетей на примере гипотезы А. Редозуба. Основные направления модернизации нейронных сетей Тестирование на компьютере по теме.

13.4.11. Перспективы использования больших данных и нейронных сетей в образовании

Основные направления использования больших данных и нейронных сетей в образовании – поиск новых методов обучения, обработка результатов тестирования, отслеживание готовности обучаемых, общее восприятие информации в учебном процессе и т.д. Тестирование на компьютере по теме.

13.4.12. Творческая работа

Выполнение творческого проекта по темам курса. Создание, обучение и использование простой нейронной сети.

13.4.13. Подведение итогов. Защита творческих проектов

Защита творческих проектов учащихся.

13.5. Выявление уровня освоения содержания

13.5.1. Требования к знаниям, умениям и способам деятельности

ЗНАТЬ

- принципы работы с большими данными;
- принцип работы нейронных искусственных сетей;

- понимать сферы применения больших данных и нейронных сетей;
- основные инструменты для работы с нейронными сетями;
- основные источники больших данных;

УМЕТЬ

- осуществлять поиск и анализ информации, необходимой для решения задач работы с большими данными и искусственными нейронными сетями;
- использовать ПО, среду программирования и различные библиотеки для обучения нейронных сетей;
- применять навыки программирования для работы с искусственной нейронной сетью;
- осуществлять поиск ошибок.

13.5.2. Использовать в своей учебной и практической деятельности

- библиотеки для работы с нейронными сетями;
- среду программирования для работы с нейронными сетями;
- организацию индивидуального информационного пространства;
- обучающие видеоресурсы и форумы специалистов;
- навыки самостоятельной и групповой деятельности;

13.6. Контрольные мероприятия

В процессе проведения элективного курса необходимо осуществлять промежуточный и итоговый контроль над процессом усваивания информации посредством проведения тестовых мероприятий и участие в творческом проекте. Кроме того, необходимо осуществлять текущий контроль посредством обратной связи от обучаемых в ходе наводящих вопросов и диалога. Необходимо строго следить, чтобы ученики не теряли «нить» в ходе получения информации. Контрольные мероприятия приведены в табл. 3.

Таблица 3

Контрольные мероприятия

№ п/п	Тема	Цель контроля	Вид контроля
1	2	3	4
2	Принцип работы с большими данными	Понимание принципов работы с большими данными	Текущий контроль. Оценка усвоения знаний.
3	Сферы применения	Понимание сфер применения больших данных	Текущий контроль. Оценка усвоения знаний.
4	Методики анализа больших данных	Освоение принципов методик анализа больших данных	Текущий контроль. Оценка усвоения знаний.
5	Примеры использования больших данных	Понимание принципов применения больших данных	Текущий контроль. Оценка усвоения знаний.
6	Понятия глубокого машинного обучения (Deep Learning) и нейронной сети	Понимание глубокого машинного обучения и связь с большими данными	Текущий контроль. Оценка усвоения знаний.
7	Архитектура нейронных сетей	Понимание архитектуры нейронных сетей	Текущий контроль. Оценка усвоения знаний.
8	Виды нейронных сетей	Понимание видов нейронных сетей	Текущий контроль. Оценка усвоения знаний.
9	Перспективы модернизации нейронных сетей	Получение представления о направлениях модернизации нейронных сетей	Текущий контроль. Оценка усвоения знаний.
10	Перспективы использования больших данных и нейронных сетей в образовании	Получение представления о перспективах использования больших данных и нейронных сетей в образовании	Текущий контроль. Оценка усвоения знаний.

1	2	3	4
11	Творческая работа.	Освоение работы с нейронными сетями - построение и обучение сети с использованием существующих библиотек.	Текущий контроль. Оценка работоспособности нейронной сети
12	Подведение итогов. Защита творческих проектов.	Освоение навыков презентации своей работы.	Итоговый контроль Оценка работоспособности нейронной сети. Оценка общего усвоения курса. Умение презентовать работу

13.7. Обоснование системы оценивания

13.7.1. Критерии оценивания усвоения темы курса

Тестирование – ответы на вопросы по темам курса.

13.7.2. Критерии оценивания результатов обучения нейронной сети

Критерии оценивания учащихся приведены в табл. 4.

Критерии оценивания учащихся

Элемент оценивания	Критерии
1	2
Алгоритм обучения нейронной сети	<ul style="list-style-type: none"> – точность построения алгоритма – наглядность алгоритма – восприятие алгоритма, аккуратность его исполнения – полнота алгоритма, отображение всех операций (достаточная детализация) – отсутствие ошибок – соответствие программному продукту – оформление алгоритма
Содержание в программном продукте, предназначенном для обучения нейронной сети	<ul style="list-style-type: none"> – соответствие программы решаемой задачи – отсутствие ошибок багов при выполнении программного кода – наличие понятных и полных комментариев в теле программного кода – оформление программного кода – наличие подключенных библиотек – умение объяснить каждой действие работы программы обучения нейронной сети

13.8. Учебно-методическое обеспечение

1. Себрант А. Что такое Big Data и почему это страшно интересно [Электронный ресурс] [2014]. URL: <https://www.youtube.com/watch?v=zsUKYfXjpvo&list=WL&index=6> (дата обращения: 04.12.2020).

2. Сербант А. Data Science, черные ящики – и почему вам сильно повезло [Электронный ресурс] [2015]. URL: <https://www.youtube.com/watch?v=zvGeLvWZ7yQ&list=WL&index=8> (дата обращения: 04.12.2020).

3. Большие данные (Big Data) [Электронный ресурс] [2017]. URL: <http://tadviser.ru/a/125096> (дата обращения: 04.12.2020).

4. Хайкин С. Нейронные сети: Полный курс. 2-ое-е изд. Издательский дом Вильямс, 2006. 1104 с.
5. Искусственная нейронная сеть [Электронный ресурс] [2018]. URL: https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C (дата обращения: 04.12.2020).
6. Редозубов А.Д. Мозг - это не нейронные сети [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=7c6YUJ0JuqI&list=WL&t=1s&index=27> (дата обращения: 04.12.2020).
7. Созыкин А. Распознавание объектов на изображениях | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=5GdtghjJ3-U&index=9&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (дата обращения: 04.12.2020).
8. Созыкин А. Введение | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=GX7qxV5nh5o&index=1&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (дата обращения: 04.12.2020).
9. Созыкин А. Сверточные нейронные сети | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=52U4BG0ENiM&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_&index=8 (дата обращения: 04.12.2020).
10. Дьячук П.П., Бортновский С.В., Дьячук П.П. Результативность бинарного и полубинарного взаимодействия обучающегося с виртуальной проблемной средой // Образовательные технологии и общество, Т. 16, № 1, 2013.

11. Дьячук П.П., Дроздова Л.Н., Шадрин И.В. Система автоматического управления учебной деятельностью и ее диагностики // Информационно-управляющие системы, № 5, 2010.

12. Ларьяновский А. Светлое будущее онлайн-образования: технологии анализа Big Data для эффективного обучения [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=WBA5MyvlBPU> (дата обращения: 04.12.2020).

ЗАКЛЮЧЕНИЕ

Таким образом, на сегодняшний день глубокое обучение нейронных сетей получает всё большее распространение практически во всех областях деятельности человека.

Скорость обработки информации практически достигла реального времени и появилась возможность работы не со статичными данными, а с постоянно прибывающим потоком. Это позволяет использовать нейронные сети там, где по естественным причинам внимание человека может быть притуплено с течением времени по естественным биологическим причинам. Также это даёт возможность исключить массу человеческих ошибок в типичных ситуациях в таких областях как медицина, различные виды транспорта, системы безопасности, экономика и финансы, социология, педагогическая деятельность и т.п.

Текущий уровень развития не даёт возможности целиком и полностью автоматизировать управление многими процессами только искусственными нейронными сетями, тем более, когда речь идёт о нестандартных ситуациях, хотя, ситуация является нестандартной, пока человек или ИНС не научатся в ней действовать.

На мой взгляд, не смотря на серьёзные продвижения в области совершенствования искусственных нейронных сетей, человечество находится только в начале этого пути. Скорее всего, будут разрабатываться новые архитектуры и способы обучения ИНС более близких к работе биологических нейронных сетей. Уже сейчас появляются отдельные мысли и работы о необходимости пересмотра имеющейся концепции искусственных сетей.

Уровень проникновения этих технологий в области обучения на сегодняшний момент является очень низким и главной причиной этому явлению в РФ, по моему мнению, является низкая коммерциализация учебной деятельности и, как следствие, недостаточность финансирования. При этом очевидно, что такие технологии, как Deep Learning и Big Data могут поднять на

новый уровень образовательный процесс и послужить толчком к улучшению существующих и выработке новых методов и методик обучения.

Также, очевидно, что внедрение машинного обучения в сфере образования поможет эффективно решать некоторые административные задачи по обработке информации большого объёма, что уже сейчас успешно применяется в других сферах деятельности человека.

Скорее всего, проникновение искусственных нейронных сетей в сферу обучения будет со стороны бизнеса, обладающего финансовыми ресурсами и стремящегося сократить расходы за счёт уменьшения времени обучения сотрудников с одновременным поднятием качества усвоения новых знаний сотрудниками компании в профессиональной деятельности, в области охраны труда и техники безопасности. Тем не менее, уже сейчас применение нейронных сетей в синхронном переводе позволяет улучшить коммуникации школьников, студентов и представителей научного сообщества из разных стран, что безусловно повышает качество образования. На мой взгляд, наивысший эффект можно получить от применения данной технологии государственными органами, т.к. позволит выработать единый системный подход. Очень важно при этом опираться на молодых специалистов, предлагающих новаторские идеи применения и конкретные технические решения.

Применение ИНС в целом и технологии Deep Learning в частности должно исключить или минимизировать таких явления, как потеря концентрации обучающихся, низкий уровень усвоения информации и даже неприязнь к отдельным дисциплинам или ко всему учебному процессу.

Становится очевидно, что эти сети будут иметь всё большее значение и участвовать в большем количестве процессов с нарастающей долей. И при дальнейшей эволюции искусственных нейронных сетей мы рано или поздно придём к созданию искусственного интеллекта.

В ходе исследования, были поставлены и решены следующие задачи:

- систематизировать и обобщить информацию по технологии глубокого машинного обучения и большими данными;
 - изучить основы теории разработки и построения элективных курсов;
 - разработать элективный курс по направлению, наполнение занятий, систему упражнений, а так же методы для текущего, промежуточного и итогового контроля над обучаемыми при прохождении элективного курса.
- Считаем, что цель нашей работы достигнута. Разработан элективный курс.

ИСТОЧНИКИ И ЛИТЕРАТУРА

1. Денисова О.О., Мухутдинов Э.А. БОЛЬШИЕ ДАННЫЕ – ЭТО НЕ ТОЛЬКО РАЗМЕР ДАННЫХ // Вестник технологического университета, Т. 18, № 4, 2015. С. 5.

2. Себрант А. Что такое Big Data и почему это страшно интересно [Электронный ресурс] [2014]. URL: <https://www.youtube.com/watch?v=zsUKYfXjpvo&list=WL&index=6> (дата обращения: 04.12.2020).

3. Сербант А. Data Science, черные ящики – и почему вам сильно повезло [Электронный ресурс] [2015]. URL: <https://www.youtube.com/watch?v=zvGeLvWZ7yQ&list=WL&index=8> (дата обращения: 04.12.2020).

4. Большие данные (Big Data) [Электронный ресурс] [2017]. URL: <http://tadviser.ru/a/125096> (дата обращения: 04.12.2020).

5. Хайкин С. Нейронные сети: Полный курс. 2-ое-е изд. Издательский дом Вильямс, 2006. 1104 с.

6. Шпаргалка по разновидностям нейронных сетей. Часть первая. Элементарные конфигурации [Электронный ресурс] [2016]. URL: <https://tproger.ru/translations/neural-network-zoo-1/> (дата обращения: 06.06.2018).

7. Искусственная нейронная сеть [Электронный ресурс] [2018]. URL: https://ru.wikipedia.org/wiki/Нейронная_сеть (дата обращения: 04.12.2020).

8. Редозубов А.Д. Мозг - это не нейронные сети [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=7c6YUJ0JuqI&list=WL&t=1s&index=27> (дата обращения: 04.12.2020).

9. Редозубов А.Д. Логика мышления. Часть 1 [Электронный ресурс] [2012]. URL: <https://www.youtube.com/watch?v=0v5OwZхох7М> (дата обращения: 04.12.2020).

10. Редозубов А.Д. Логика сознания. Вступление [Электронный ресурс] [2016]. URL: <https://habrahabr.ru/post/308268/> (дата обращения: 04.12.2020).
11. Редозубов А.Д. Логика эмоций. 2012. 317 с.
12. Развина Э. Вспомнить все. А если это нереально? [Электронный ресурс] [2014]. URL: <https://www.b17.ru/blog/15224/> (дата обращения: 04.12.2020).
13. Созыкин А. Распознавание объектов на изображениях | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=5GdtghjJ3-U&index=9&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (дата обращения: 04.12.2020).
14. Созыкин А. Введение | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=GX7qxV5nh5o&index=1&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (дата обращения: 04.12.2020).
15. Созыкин А. Сверточные нейронные сети | Глубокие нейронные сети на Python [Электронный ресурс] [2016]. URL: https://www.youtube.com/watch?v=52U4BG0ENiM&list=PLtPJ9IKvJ4oiz9aaL_xcZd-x0qd8G0VN_&index=8 (дата обращения: 06.06.2018).
16. Шумахер М. | TensorFlow.js и clmtrackr.js: отслеживание направления взгляда пользователя в браузере [Электронный ресурс] [2018]. URL: <https://habr.com/ru/company/ruvds/blog/426055/> (дата обращения: 04.12.2020).
17. Куда расти в ИТ: три самые продвинутые специализации 14 04 2017. [Электронный ресурс]. URL: <https://krasnoyarsk.hh.ru/article/305220>. (дата обращения: 04.12.2020).
18. Канеман Д. Думай медленно. Решай быстро. 2013. 656 с.
19. Дьячук П.П., Бортновский С.В., Дьячук П.П. Результативность бинарного и полубинарного взаимодействия обучающегося с виртуальной проблемной средой // Образовательные технологии и общество, Т. 16, № 1, 2013.

20. Дьячук П.П., Дроздова Л.Н., Шадрин И.В. Система автоматического управления учебной деятельностью и ее диагностики // Информационно-управляющие системы, № 5, 2010.

21. Дроздова Л.Н., Дьячук П.П., Кудрявцев В.С. Диагностика динамических параметров учебной деятельности по конструированию звуковых объектов // Системы управления и информационные технологии, Т. 39, № 1.2, 2010. С. 233-238.

22. Волков А.И. Труды Международной научной конференции // Проблемы интеграции хранилищ данных с открытыми и большими данными и подходы к их решению. 2015. С. 18-32.

23. Арустамов А. Анализ больших объемов данных [Электронный ресурс] URL: <https://basegroup.ru/community/articles/very-large-data> (дата обращения: 06.06.2018).

24. BigData и машинное обучение для e-commerce и не только [Электронный ресурс] [2015]. URL: <https://www.youtube.com/watch?v=tywGoQv1G6I&list=WL&index=7> (дата обращения: 04.12.2020).

25. Краткий обзор популярных нейронных сетей [Электронный ресурс] [2010]. URL: <https://geektimes.ru/post/83781/> (дата обращения: 06.06.2018).

26. Работа с Big Data: основные области и возможности [Электронный ресурс] [2014]. URL: http://www.marketing.spb.ru/lib-research/methods/Big_Data.htm (дата обращения: 04.12.2020).

27. Революция Big Data: Как извлечь необходимую информацию из "Больших Данных"? [Электронный ресурс] URL: <http://www.statsoft.ru/products/Enterprise/big-data.php>

28. Давлеткалиев Р. Что такое свёрточная нейронная сеть [Электронный ресурс] [2016]. URL: <https://habrahabr.ru/post/309508/> (дата обращения: 04.12.2020).

29. Ефремова Н. Нейронные сети: практическое применение [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=dCf9cyDHzeI> (дата обращения: 04.12.2020).

30. Коробченко Д. Лекция Дмитрия Коробченко по Deep Learning [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=ytRumlP4NAo&list=WL&t=5121s&index=18> (дата обращения: 04.12.2020).

31. Ларьяновский А. Светлое будущее онлайн-образования: технологии анализа Big Data для эффективного обучения [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=WBA5MyvlBPU> (дата обращения: 04.12.2020).

32. Петров А. Лекция Дмитрия Коробченко по Машинному Обучению и Deep Learning [Электронный ресурс] [2016]. URL: <https://www.youtube.com/watch?v=6jbU5gLNzz4&t=111s> (дата обращения: 04.12.2020).

33. Сиротенко М.Ю. Применение нейросетей в распознавании изображений [Электронный ресурс] [2009]. URL: <https://geektimes.ru/post/74326/> (дата обращения: 04.12.2020).