федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 Программная инженерия
Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка и применение нейронной сети для интеллектуального контент-анализа социальных сетей

УДК 004.414.2:316.472:004.891.3

Студент

Группа	ФИО	Подпись	Дата
8К4Б	Чудин Игорь		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГТ	Петухов Олег Николаевич	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОКД	Авдеева Ирина Ивановна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Евгений Сергеевич	к.т.н.		

Томск – 2018 г.

Планируемые результаты обучения по ООП

Код результатов	Результат обучения (выпускник должен быть готов)
Р1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
Р2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
Р3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с применением базовых и специальных знаний, современных аналитических методов и моделей.
Р4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с применением средств автоматизации проектирования.
Р5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
Р6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Р7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
Р8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.

Р9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
Р10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
Р11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Школа ИШИТР
Направление подготовки 09.03.04 Программная инженерия
Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

_____ _____ _____
(Подпись) (Дата)
(Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	**ФИО**
8К4Б	Чудину Игорю

Тема работы:

Разработка и применение нейронной сети для интеллектуального контент-анализа социальных сетей	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Работа направлена на создание модуля для тонального анализа текста на основании нейронных сетей,

	сообщающая пользователю оценку тональности сообщения
Перечень подлежащих исследованию, проектированию и разработке вопросов	Исследование методов классификации, основанных на нейронных сетях, и классических (линейных и нелинейных) методов; Разработка свёрточной нейронной сети, рекуррентной нейронной сети и их обучение с использованием Tensorflow; Оценка точности разработанных алгоритмов и их сравнение Проектирование и реализация веб модуля для тонального анализа текста. Расчет ресурсоэффективности и ресурсосбережения. Анализ вредных производственных факторов.
Перечень графического материала *(с точным указанием обязательных чертежей)*	презентация

Консультанты по разделам выпускной квалификационной работы

Раздел	**Консультант**
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент отделения ОСГН Петухов Олег Николаевич
Социальная ответственность	Ассистент отделения ОКД Авдеева Ирина Ивановна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4Б	Чудин Игорь		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Школа информационных технологий и робототехники
Направление подготовки (специальность) 09.03.04 Программная инженерия
Уровень образования Бакалавриат
Отделение школы (НОЦ) Информационных технологий
Период выполнения осенний / весенний семестр 2017/2018 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.03.2018	Раздел 1. Обзор предметной области	20
17.03.2018	Раздел 2. Проектирование и теоретическое обоснование системы	35
26.05.2018	Раздел 3. Реализация системы	35
20.05.2018	Раздел 4:. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
25.05.2018	Раздел 5. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Евгений Сергеевич	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа 128 с., 24 рис., 29 табл., 17 источников, 1 прил.

Ключевые слова: искусственные нейронные сети, глубокие нейронные сети, обучение с учителем, глубокое обучение, рекуррентная нейронная сеть, LSTM, свёрточная нейронная сеть, анализ тональности текста, мешок слов, word2vec.

Объектом исследования являются методы на основе нейронных сетей для анализа тональности корпуса текстов.

Для достижения поставленной в работе цели необходимо решить следующие задачи:

- Изучить теоретический материал про обучение глубинных нейронных сетей и их особенности применительно к обработке естественного языка;
- Изучить документацию библиотеки Tensorflow;
- Разработать модели свёрточной и рекуррентной нейронных сетей;
- Разработать реализацию линейных и нелинейных методов классификации на моделях мешка слов и Word2Vec;
- Сравнить точность и другие показатели качества реализованных нейросетевых моделей с классическими методами.

Для визуализации обучения используется Tensorboard.

В работе показано преимущество классификаторов на основе глубоких нейронных сетей над классическими методами классификации, даже если для векторных представлений слов используется модель Word2Vec. Самую высокую предсказательную точность для данного корпуса текстов имеет модель рекуррентной нейронной сети с LSTM-блоками.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

Deep learning — глубокое обучение;

Supervised learning — обучение с учителем;

SVM (support vector machine) — метод опорных векторов;

Pooling — операция объединения в свёрточных нейронных сетях;

Softmax — функция мягкого максимума;

Dropout — метод регуляризации для предотвращения переобучения сети;

Batches — группы примеров, используемые для обучения сети;

CBOW (continuous bag of words) — непрерывный мешок со словами, один из алгоритмов обучения Word2Vec;

CNN (convolutional neural network)— свёрточная нейронная сеть;

RNN (recurrent neural network) — рекуррентная нейронная сеть;

LSTM (long short term memory) — долгая краткосрочная память, разновидность архитектуры рекуррентных нейронных сетей.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современный мир нельзя представить без чтения и обработки информации. Объем информации, которую получает человек, растёт в огромном количестве. И эта информация может быть обработана различными информационными системами. На текущее время самый простой для инженера-программиста способ, это нейросеть. Нейросетями обрабатывается любая информация, от графической до огромных массивов данных. В данной работе мы рассматриваем применение нейросети для анализа тональности текста.

Ежедневно в сети возникает большое количество различного контента: тысячи комментариев пользователей социальных сетей и форумов на различные темы, например, такие как политика, спорт, развлечения, работа, развитие технологий, международные отношения и так далее. И каждый выражает свою точку зрения, отзывы и оценки. Благодаря этому возникает вопрос анализа эмоциональной окраски текста. Для её решения используются методы анализа тональности текста. Решение данной проблемы достаточно известная задача.

Анализ тональности текста помогает научить информационную систему воспринимать естественный язык, а также использовать и применять естественный язык на уровне, похожем на человеческий. Такой анализ способен повысить качество машинного перевода, научить машину «думать, как переводчик», приняв во внимание все те соображения, которыми пользуется профессионал переводчик, а также поможет определить мнение автора текста.

В ходе работы нужно было построить бинарный классификатор, определяющий каким, оказался текст, для этого были выбраны три характеристики позитивная, негативная и нейтральная. Для достижения

данной цели были использованы различные способы и векторные модели представлений.

В результате обучения были получены модели нейронных сетей, позволяющие с достаточной достоверностью определять тональность текста, а также проведено сравнение эффективности использования реализованных методов. Конечным результатом стала проверка системы на тестовой выборке самым точным методом.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1.Понятие нейронной сети

Искусственные нейронные сети и их составляющие

Искусственные нейронные сети были построены по принципу биологических нейронных сетей, которые представляют собой сети нервных клеток, выполняющие определенные физиологические функции. Составным элементом нейронных сетей являются нейроны (представлены на рис.1.1).

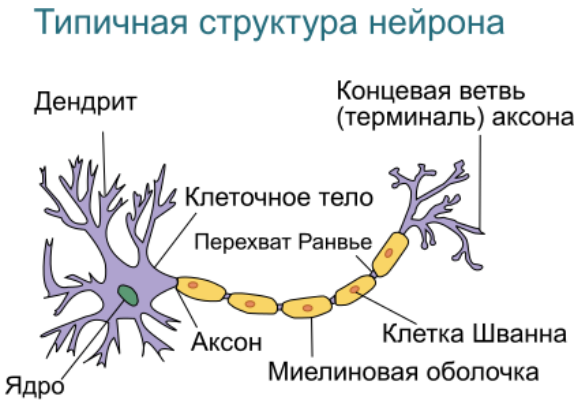

Рисунок 1.1 — Типичная структура нейрона

У нейрона есть несколько функций:

- Приёмная функция: синапсы получают информацию;
- Интегративная функция: на выходе нейрона сигнал, который несёт информацию обо всех суммированных в нейроне сигналах;
- Проводниковая функция: по аксону проходит информация к синапсу;
- Передающая функция: импульс, достигший окончания аксона, заставляет медиатор передавать возбуждение следующему нейрону.

Синапсами называют связи, по которым выходные сигналы одних нейронов поступают на входы других. Каждая связь характеризуется своим весом. Связи с положительным весом называются возбуждающими, а с отрицательным — тормозящими. Выход нейрона называется аксоном. В

искусственной нейронной сети искусственный нейрон — это некоторая нелинейная функция, аргументом которой является линейная комбинация всех входных сигналов. Такая функция называется активационной. Затем результат активационной функции посылается на выход нейрона. Объединяя такие нейроны с другими, получают искусственную нейронную сеть.

Активационная функция

Функция активации нейрона характеризует зависимость сигнала на выходе нейрона от суммы сигналов на его входах. Обычно функция является монотонно возрастающей и находится в области значений [-1,1] (гиперболический тангенс) и [0,1] (сигмоида). Для некоторых алгоритмов обучения необходимо, чтобы активационная функция была непрерывно дифференцируемой на всей числовой оси. Искусственный нейрон характеризуется своей активационной функцией (например, название "сигмоидальный нейрон").

Основными активационными функциями являются:

Пороговая активационная функция (функция Хевисайда). Нельзя использовать для алгоритма обратного распространения ошибки;

$$f(x) = \begin{cases} 1, & x \geq -\omega 0 x 0 \\ 0, & else \end{cases} \qquad (1.1)$$

• Сигмоидальная активационная функция;

$$\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}} \qquad (1.2)$$

• Гиперболический тангенс.

$$\tanh(\mathrm{A}x) = \frac{e^{\mathrm{A}x} - e^{-\mathrm{A}x}}{e^{\mathrm{A}x} + e^{-\mathrm{A}x}} \qquad (1.3)$$

Перцептрон

Перцептрон — тип искусственного нейрона, разрабатываемый Фрэнком Розенблаттом в 1950-ых и 1960-ых годах. В современных работах чаще всего используют другую модель искуственного нейрона —

сигмоидальный нейрон. Чтобы понять, как работает сигмоидальный нейрон, необходимо

рассмотреть структуру и принцип работы перцептрона. Перцептрон принимает на вход значения $x1$, $x2$... и выдаёт бинарный результат (см. рис.3.2).

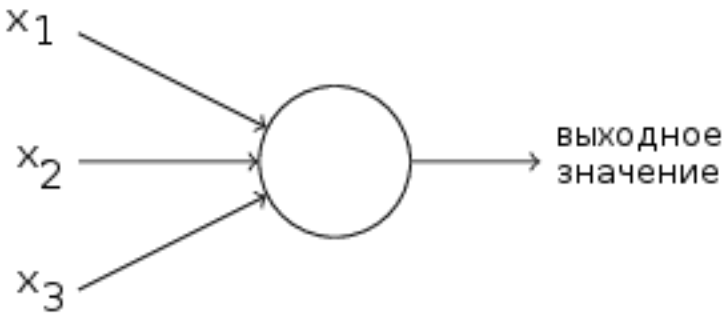

Рисунок 1.2 — Схема перцептрона

Розенблатт предложил использовать веса — числа, выражающие важность вклада каждого входа в конечный результат. Взвешенная сумма (или веса) сравнивается с пороговым значением (threshold), и по результатам определяется, будет ли выдан 0 или 1. Пороговое значение также является параметром нейрона.

$$\begin{cases} 0, \ IF \ \sum_j \omega 0 x 0 \ \leq threshold \\ 1, \ IF \ \sum_j \omega 0 x 0 \ > threshold \end{cases} \quad (1.4)$$

Перцептроны могут быть классифицированы как искусственные нейронные сети

- с одним скрытым слоем;
- с пороговой активационной функцией;
- с прямым распространением сигнала.

Обучение перцептрона состоит в изменении матрицы весов в процессе обучения. Существуют 4 исторически сложившихся видов перцептронов:

- Перцептрон с одним скрытым слоем;

- Однослойный перцептрон: входные элементы напрямую соединены с выходными с помощью системы весов. Является простейшей сетью прямого распространения (feedforward network);
- Многослойный перцептрон (по Розенблатту): присутствуют дополнительные скрытые слои;
- Многослойный перцептрон (по Румельхарту): присутствуют дополнительные скрытые слои, а обучение проводится по методу обратного распространения ошибки (backpropagation algorithm).

Если бы небольшое изменение весов (или смещения) вызывало небольшое же изменение на выходе сети, то желаемое поведение нейронной сети можно было бы получить с помощью простых модификаций смещений и весов в процессе обучения. Однако обучение не так просто осуществить, если нейронная сеть состоит из перцептронов. Небольшое изменение весов или смещения одного из перцептронов сети может кардинально изменить выходное значение перцептрона, например, с 0 на 1. Поэтому самое незначительное изменение значений одного из элементов сети может создать значительные трудности в понимании изменения поведения сети. Поскольку задача обучения нейронной сети является задачей поиска минимума функции ошибки в пространстве состояний обучения, то для ее решения могут применяться стандартные методы теории оптимизации. Для однослойного перцептрона с n входами и m выходами речь идет о поиске минимума в nm-мерном пространстве.

Сигмоидальный нейрон

Сигмоидальные нейроны похожи на перцептроны, однако небольшие изменения в их весах и смещениях незначительно изменяют выход нейрона.

Этот факт позволяет сети из сигмоидальных нейронов обучаться. На вход сигмоидального нейрона подаются любые значения между 0 и 1. На выходе также выдаётся значение между 0 и 1, так как в качестве

активационной функции используется сигмоида, являющаяся нелинейной (см. рис. 1.3).

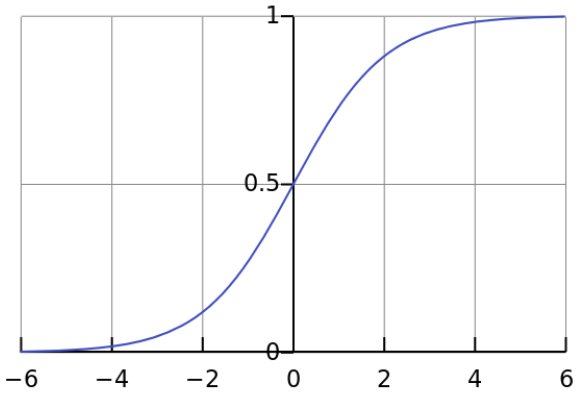

Рисунок 1.3 — График логистической кривой сигмоиды

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

Чем больше β (параметр наклона сигмоидальной функции активации), тем сильнее крутизна графика. При $\beta \to \infty$ сигмоида стремится к функции Хевисайда.

Важным свойством сигмоидальной функции является её дифференцируемость. Применение непрерывной функции активации позволяет использовать при обучении градиентные методы.

Нейроны можно разделить на группы в зависимости от их положения в сети:

- входные нейроны принимают исходный вектор данных;
- в промежуточных нейронах происходят основные вычислительные операции — обучение;
- выходные нейроны — результат работы сети.

Архитектура нейронных сетей

Рассмотрим задачу обучения с учителем. Дано множество тренировочных примеров X с метками (labels) Y. Нейронные сети определяют нелинейную гипотезу hW,b(x) с параметрами W и b. Нейронная сеть

составлена из множества простых нейронов так, что выход одного из нейронов будет входом другого (см. рис. 1.4).

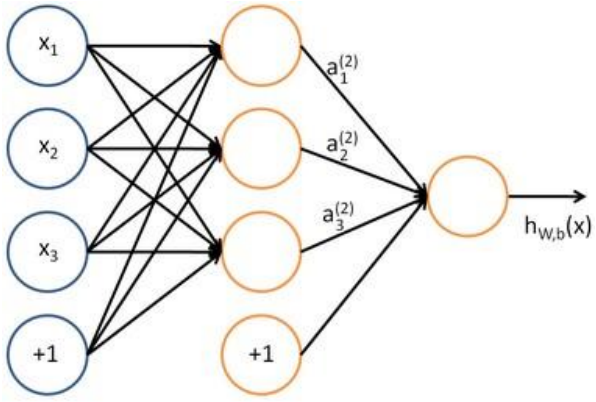

Рисунок 1.4 — Схема простейшей сети прямого распространения

Крайний левый слой называется входным, а крайний правый слой — выходным. Слой посередине является скрытым и называется так из-за того, что его значения не наблюдаются в тренировочных примерах. Таким образом в данной сети три элемента входа, три скрытых элемента и один выходной элемент. Пусть nl — количество слоёв в сети (в данном случае 3). Параметры сети (W, b) = (W(1), b(1), W(2), b(2)).

Результат применения функции активации (выхода) обозначается ai для

i-ого элемента. Получаем такую систему:

$$\begin{cases} a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_{(1)}^1 \\ a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_{(1)}^2 \end{cases} \quad (1.6)$$

Обозначив функцию суммирования через z, получим в векторной форме:

$$\begin{cases} z^{(2)} = W^{(1)}x + b^{(1)} \\ a^{(2)} = f(z^{(2)}) \\ z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \\ h = f(z^{(3)}) \end{cases} \quad (1.7)$$

Общая формула будет выглядеть таким образом:

$$\begin{cases} z^{(l)} = W^{(l)}a^{(l)} + b^{(l)} \\ a^{(l)} = f(z^{(l+1)}) \end{cases} \quad (3.8)$$

Сетью прямого распространения называются нейронные сети, которые используют выход одного слоя в качестве входных данных для следующего слоя.

Обучение нейронных сетей

Общие понятия в обучении нейронных сетей

Эпоха – прямой и обратный проход по всем тренировочным примерам.

Размер серии (batch) – количество тренировочных примеров для одной итерации прямого и обратного проходов.

Количество итераций – количество проходов: каждый проход использует примеры (batch). Один проход = прямой проход + обратный проход.

То есть имея 1000 примеров, batch = 500, нам потребуется две итерации, чтобы завершить одну эпоху.

С математической точки зрения, обучение нейронных сетей – многопараметрическая задача нелинейной оптимизации.

Алгоритм обратного распространения ошибок

Алгоритм обратного распространения ошибки определяет стратегию

подбора весов многослойной сети с применением градиентных методов оптимизации. Поскольку целевая функция, обычно определяемая как квадратичная разность суммы между фактическими и ожидаемыми выходными значениями, является непрерывной, градиентные методы оптимизации являются эффективными при обучении сети. При обучении

многослойной нейронной сети необходимо вычислить вектор градиента относительно параметров всех слоёв сети. Пусть имеется конечный набор тренировочных данных (m примеров). Для обучения нейронной сети применяют пакетный градиентный спуск (batch gradient descent). Квадратичная ошибка целевой функции (squared-error cost function) для одного примера будет вычислена по формуле:

$$J(W,b,x,y) = \frac{1}{2}\left\|h_{W,b}(x) - y\right\|^2 \qquad (3.9)$$

Тогда целевая функция для m примеров будет выглядеть так:

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b,\,x^i,\,y^i\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ij}^{(l)})^2 =$$

$$\left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^i\right) - y^i\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ij}^{(l)})^2 \qquad (3.10)$$

Первый член выражения J(W,b) это сумма квадратов ошибок, второй — член регуляризации (L2 — уменьшение весов — weight decay), позволяющий уменьшить значения весов и предотвратить переобучение. Параметр регуляризации весов λ используют для проверки относительной значимости частей данного выражения. В задачах бинарной классификации y представлен 0 или 1 (так как сигмоидная функция выдает значение в пределах [0;1]; однако при использовании гиперболического тангенса лейблами классов были бы -1 и 1). Задача — минимизировать J(W,b). Для обучения нейронной сети необходимо инициализировать каждый параметр $W_{ij}^{(l)}$ и $b_i^{(l)}$ малыми случайными величинами, близкими к нулю, а затем применить алгоритм оптимизации (упоминавшийся выше градиентный спуск).

Так как J(W, b) не является выпуклой функцией, то градиентный спуск восприимчив к локальным оптимумам. Каждая итерация градиентного спуска обновляет параметры таким образом:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial}{\partial W_{ij}^{(l)}}J(W,b)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \qquad (3.11)$$

где α — скорость обучения (learning rate).

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) = \left[\frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial W_{ij}^{(l)}} J\left(W, b, x^i, y^i\right) \right] + \lambda W_{ij}^{(l)} \qquad (1.12)$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial b_i^{(l)}} J\left(W, b, x^i, y^i\right) \qquad (1.13)$$

Шаги алгоритма обратного распространения ошибки

1) Осуществляется прямой проход по сети, вычисляются активации слоёв L2, L3 и так далее до выходного слоя Lnl;

2) Для каждого выходного элемента i в выходном слое nl (the output layer) рассчитывается ошибка

$$\delta_i^l = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \left\| h_{W,b}(x^i) - y^i \right\|^2 = -\left(y_i - a_i^{n_i}\right) \cdot f'(z_i^{n_i}) \qquad (1.14)$$

3) Для l = nl − 1, nl − 2, nl − 3, . . . , 2:

Для каждого элемента в слое l, рассчитывается

$$\delta_i^l = \left(\sum_{j=1}^{s_{l+1}} W_{ij}^{(l)} \delta_i^{l+1} \right) f'(z_i^l) \qquad (1.15)$$

4) Вычисляются частные производные

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b, x, y) = a_j^l \delta_i^{l+1} \qquad (1.16)$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b, x, y) = \delta_i^{l+1} \qquad (1.17)$$

В матричной форме алгоритм будет записан таким образом (• обозначено произведение Адамара — покомпонентное произведение двух матриц):

1) Осуществляется прямой проход по сети, вычисляются активации слоёв L2, L3 и так далее до выходного слоя Lnl

2) Матрица ошибок для выходного слоя nl

$$\delta^{n_l} = -(y_i - a^{n_i}) \cdot f'(z^{n_i}) \qquad (1.18)$$

3) Для слоя l = nl − 1, nl − 2, nl − 3, . . . , 2

$$\delta^l = ((W^l)^T \delta^{l+1}) \cdot f'(z^l) \quad (1.19)$$

4) Вычисление частных производных

$$. \nabla_{w^l} J(W, b, x, y) = \delta_i^{l+1} (a^l)^T \quad (1.20)$$

$$. \nabla_{b^l} J(W, b, x, y) = \delta^{l+1} \quad (1.21)$$

Недостатки градиентного спуска

Основная трудность обучения нейронных сетей состоит в методах выхода из локальных минимумов. Недостатками градиентного спуска при обучении сети являются:

• Паралич сети

Значения весов сети в результате коррекции могут стать очень большими величинами. Поскольку ошибка, посылаемая обратно в процессе обучения, пропорциональна производной сжимающей функции, то процесс обучения может почти остановиться. Это можно предотвратить, уменьшая шаг η, одна процесс обучения будет происходить дольше.

• Размер шага

Если значение шага не изменяется, и оно довольно мало, то метод сходиться слишком медленно. Если же шаг слишком велик, то может возникнуть паралич сети. Необходимо изменять значение шага: увеличивать до тех пор, пока не прекратится улучшение оценки в направление антиградиента и уменьшать, если оценка не улучшается.

Сравнение стохастического и пакетного градиентных спусков

Если для пакетного градиентного спуска функция потерь вычисляется для всех образцов вместе взятых после окончания эпохи, а потом изменяются весовые коэффициенты нейронов, то для стохастического метода весовые коэффициенты изменяются после вычисления выхода сети на одном из обучающих примеров. Недостатком пакетного градиентного спуска является его “застревание” в локальных минимумах. Несмотря на то, что

стохастический метод работает медленнее пакетного, он способен выходить из локальных минимумов, что приводит к лучшим результатам обучения сети (стохастический метод использует недовычисленный градиент).

Мониторинг состояния сети

Функция перекрёстной энтропии в качестве целевой функции

Функция перекрёстной энтропии используется в качестве функции потерь: y_i'— предсказанные значения, yi — верные значения.

$$L(x, y) = -\frac{1}{n}\sum_{i=1}^{n} y^i \log a\left(x^i\right) + \left(1 - y^i\right) \log 1 - a\left(x^i\right) \qquad (1.22)$$

Техники регуляризации

• L1-регуляризация: происходит изменение нерегуляризованной целевой функции путём добавления суммы абсолютных значений весов:

$$J = J_0 + \frac{\lambda}{n}\sum_W |W| \qquad (1.23)$$

При использовании L1-регуляризации происходит стремление одного

или более весовых значений к 0.0, поэтому соответствующая функция (feature) больше не требуется. Этот эффект называется селекцией функций (feature selection);

• L2-регуляризация (также известная как weight decay) В отличии от L1-регуляризации, в L2 веса уменьшаются на величину, пропорциональную весам:

$$J = J_0 + \frac{\lambda}{2n}\sum_W W^2 \qquad (1.24)$$

• Dropout не влияет на значение целевой функции: изменяется структура сети. Каждый нейрон удаляется из сети с некоторой вероятностью p. По полученной прореженной сети делается обратное распространение ошибки, для оставшихся весов делается градиентный шаг. После этого удалённые нейроны восстанавливают в сети. При обучении нейросети выход каждого нейрона домножается на (1-p). Так будет получено матожидание ответа сети по её 2N (где N — количество нейронов в сети) архитектурам.

Обученная таким образом сеть является результатом усреднения 2N сетей. Отдельная нейронная сеть, обученная при помощи раннего останова, имеет слишком большую ошибку, однако усреднение нескольких нейронных сетей приводит к существенному снижению ошибки;

• Искусственное расширение данных для обучения.

Гиперпараметры сети

Темп обучения: сначала необходимо оценить пороговое значение для η, в котором значение целевой функции мгновенно начинает снижаться без колебаний. Сначала значение оценки устанавливается η = 0.01. Если значение целевой функций снижается во время первых эпох, то нужно увеличивать темп обучения, пока будет не найдено значение колебания целевой функции. Если же при начальном темпе обучения значения целевой функции колеблются, то необходимо его уменьшать. Темп обучения регулирует размер шага в градиентном спуске и наблюдает за значениями целевой функции, определяя, был ли размер шага градиентного спуска слишком большим;

Использовать раннюю остановку (early stopping) для определения размера эпох обучения: ранняя остановка значит, что в конце каждой эпохи нужно вычислить достоверность классификации на данных проверки (validation set). Когда улучшение точности прекратится, остановить процесс обучения. Такая остановка также предотвращает переобучение;

График обучения: идея — сохранять темп обучения неизменным до момента, когда достоверность данных проверки не начнёт ухудшаться. Тогда необходимо уменьшить темп обучения (уменьшив, например, в 10раз).

Параметр регуляризации λ: после определения η, можно начать с λ=1.0 и затем увеличивать или уменьшать значение (в 10 раз);

Размер пакетов: если размер пакетов слишком мал, невозможно полностью использовать преимущества хороших матричных библиотек, оптимизированных для быстрого оборудования. Если же размер пакетов

слишком велик, то веса сети будут обновлять очень нечасто. Необходимо выбрать компромиссное значение, которое максимизирует скорость обучения.

Глубокие нейронные сети

Глубокими нейронными сетями называются такие сети, в которых есть несколько скрытых слоев. Поскольку каждый скрытый слой вычисляет нелинейное преобразование предыдущего слоя, глубокая сеть может иметь значительно большую репрезентативную мощность (то есть может представлять значительно более сложные функции), чем малослойная. При обучении глубокой сети важно использовать нелинейную функцию активации в каждом скрытом слое. Это связано с тем, что множество слоев линейных функций сами вычисляли бы только линейную функцию ввода и, следовательно, не были бы более выразительными, чем применение только одного скрытого слоя.

Главным достоинством глубинных сетей является сжатое представление достаточного большого множества функций. Можно показать, что существуют функции, которые k-слойная сеть может представлять сжато, а (k-1)-слойная сеть не может этого сделать, если только она не имеет экспоненциально большое количество элементов в скрытых слоях.

Доступность данных

С помощью метода, описанного выше, можно полагаться только на маркированные данные для обучения. Однако помеченных данных часто бывают недостаточно, и, следовательно, для многих задач трудно получить достаточное количество примеров для соответствия параметрам сложной модели. Например, учитывая высокую степень выразительности глубинных сетей, обучение при небольшом количестве данных приведет к переобучению.

Локальный оптимум

Обучение малослойной сети (с 1 скрытым слоем) с применением контролируемого обучения обычно приводит к сближению параметров с

подходящими значениями. Но при обучении глубокой сети, это работает намного реже. В частности, обучение нейронной сети с применением обучения с учителем включает в себя решение проблемы с невыпуклой оптимизацией (например, минимизация ошибки обучения $\sum_i \left\| h_{W,b}(x^i) - y^i \right\|^2$ зависимости от сетевых параметров W). В глубокой сети появляется большое количество локальных оптимумов, поэтому обучение с градиентным спуском перестаёт работать.

Градиентная диффузия

При использовании метода обратного распространения ошибки для вычисления производных, градиенты, которые распространяются от выходного слоя до более ранних слоев сети, быстро уменьшаются по мере увеличения глубины сети. В результате производная от общей стоимости по отношению к весам в более ранних слоях очень мала. Таким образом, при использовании градиентного спуска веса ранних слоев медленно меняются и более ранние слои не могут многому научиться. Эту проблему часто называют “диффузией градиентов” (diffusion of gradients).

1.2. Проблемы обучения глубоких сетей и их решения

Исчезающий градиент

Проблема исчезающего градиента — это трудность, возникающая при обучении искусственных нейронных сетей с применением методов обучения на основе градиента и обратного распространения ошибки. В таких методах каждый вес нейронной сети обновляется пропорционально градиенту функции ошибки относительно текущего веса на каждой итерации обучения. Стандартные функции активации, такие как гиперболический тангенс, имеют градиенты в диапазоне (-1, 1), а метод обратного распространения ошибки вычисляет их по цепному правилу. После умножения этих чисел для вычисления градиентов “фронтальных” слоев в n-слойной сети, что означает,

что градиент (сигнал ошибки) экспоненциально уменьшается вместе с n, а передние слои обучаются очень медленно. Когда используются функции активации, производные которых могут принимать большие значения, есть риск столкнуться с exploding gradient problem. Возможными решениями являются:

Многоуровневая иерархия: слой сети предварительно обучается, используя методы обучения без учителя, а затем его значение регулируется с помощью метода обратного распространения ошибки. Таким образом каждый слой сети изучает сжатое представление наблюдений, которое подается на следующий слой;

Долгая краткосрочная память: разновидность архитектуры рекуррентных нейронных сетей. Когда величины ошибки распространяются в обратном направлении от выходного слоя, ошибка не выпускается из памяти LSTM-блока. Она непрерывно передаётся обратно каждому из вентилей, пока они не будут обучены отбрасывать подобные значения

Остаточные сети (Residual networks): один из наиболее эффективных методов решения проблемы исчезающего градиента является применение остаточных нейронных сетей (ResNets). Более глубокая сеть будет иметь более высокую ошибку обучения, чем малослойная сеть. Команда Microsoft Research обнаружила, что разделение глубокой сети на части (скажем, каждый часть представляет собой три слоя сети) и передача входных данных в каждый фрагмент до следующего фрагмента (наряду с остаточным выходом Из куска минус входные данные вновь введённого фрагмента) помогли устранить большую часть этой проблемы с исчезновением градиента. Никаких дополнительных параметров или изменений в алгоритме обучения не требуется. ResNets показали более низкую ошибку обучения (и тестовую ошибку), чем их более малослойные аналоги, путем повторного ввода выходов из более мелких слоев в сети для компенсации исчезающих данных.

Сигмоидальные активационные функции

Применение сигмоидальных активационных функций может вызвать проблемы в обучении глубинных сетей, а именно значения активаций в конечном слое будут близки к нулю на ранних этапах обучения, замедляя этот процесс. Были предложены альтернативные активационные функции, которые не так страдают от ограничения.

Выбор подходящих весов

Выбор подходящих весов и momentum schedule в импульсном стохастическом градиентном спуске (momentum-based stochastic gradient descent) существенно влияют на способность обучать глубокие сети.

Свёрточные нейронные сети

Свёртка является операцией, которая применяется к двум последовательностям f и g и порождает третью последовательность.

$$(f * g)(c) = \sum_a f(a)\, g(c-a), \text{где } a = b + c \qquad (1.25)$$

Формула для двумерной свёртки:

$$(f * g)(c_1, c_2) = \sum_a f(a_1, a_2)\, g(c_1 - a_1, c_2 - a_2) \qquad (1.26)$$

Рассмотрим одномерный свёрточный слой с входами xn и выходами yn (см. рис. 1.5). Тогда функция для выходов будет представлена следующим образом:

$$y_n = A(x_n, x_{n+1} \ldots) \qquad (1.27)$$

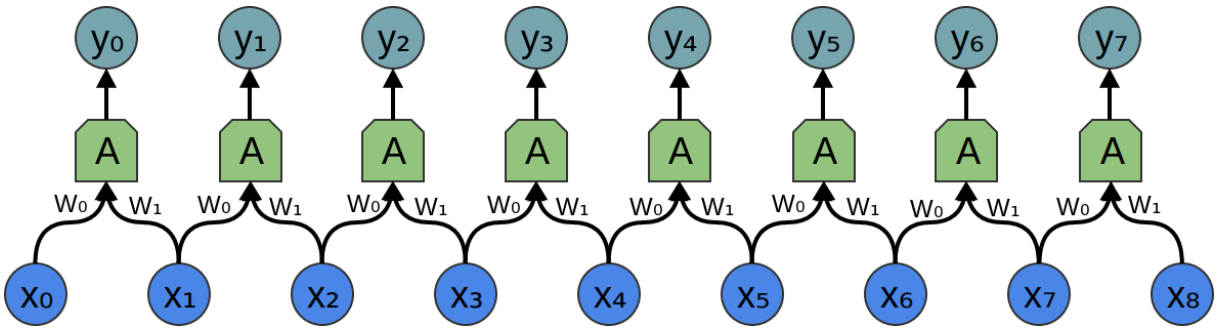

Рисунок 1.5 — Пример одномерного свёрточного слоя

В свёрточном слое находится множества копий одного и того же нейрона, поэтому многие веса появляются в нескольких позициях.

y0 = σ(W0x0 + W1x1 − b) (1.28)

y1 = σ(W0x1 + W1x2 − b) (1.29)

Стандартная матрица весов соединяет каждый вход с каждым нейроном с разными весами. Матрица для свёрточного слоя отличается тем, что различные веса могут появляться на нескольких позициях, а поскольку нейроны не соединены со всеми возможными входами, матрица содержит множество нулевых элементов:

$$M = \begin{bmatrix} \omega_0 & \omega_1 & 0 & \dots \\ 0 & \omega_0 & \omega_1 & \dots \\ 0 & 0 & \omega_0 & \dots \end{bmatrix} \qquad (1.30)$$

То есть умножение на матрицу выше — то же самое, что и свёртка с [. . . 0, w1, w0, 0 . . .]. Ядро свёртки, скользящее по разным частям изображения, соответствует наличию нейронов в этих частях.

Свёртку можно пояснить на примере обработки изображений. Если представить, что изображения – двумерные функции, то различные преобразования изображений не что иное, как свёртка функции изображения с локальной функцией, которая называется ядром свёртки.

Каждый новый пиксель изображения представляет собой взвешенную сумму пикселей, которые ядро прошло к этому моменту времени. Двумерный свёрточный слой представлен на рис. 1.6.

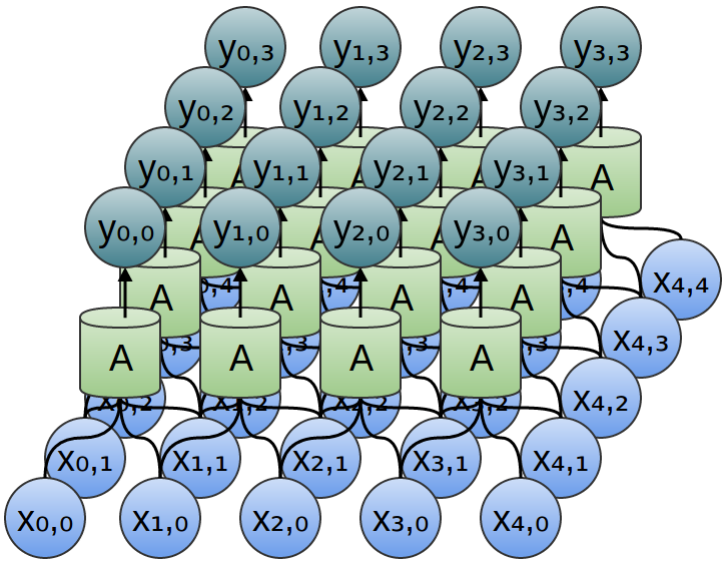

Рисунок 1.6 — Двумерный свёрточный слой

Свёрточная нейронная сеть — архитектура нейронных сетей, изначально созданная и использованная для эффективного распознавания изображений: чередуются свёрточные слои (convolutions) с нелинейными активационными функциями (ReLU или гиперболический тангенс tanh) и слои объединения (pooling layers).

В отличии от сети прямого распространения, где каждый входной нейрон соединяется с выходным нейроном в следующем слое, в свёрточных сетях для получения выходных значений используются свёртки над каждым входным слоем. В операции свёртки используется матрица весов небольшого размера, которая сдвигается по всему обрабатываемому слою, формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. Эта матрица называется ядром свёртки; она используется для различных нейронов выходного слоя.

При выполнении операции свёртки каждый фрагмент (например, изображения) поэлементно умножается на матрицу свёртки, а результат суммируется и записывается в аналогичную позицию выходного изображения. Матрицу свёртки представляет собой графическое кодирование какого- либо признака. Получившийся в результате операции свёртки следующий слой

показывает наличие данного признака. В свёрточной нейронной сети существуют много наборов весов, которые кодируют элементы изображений. Ядра свёртки формируются в процессе обучения сети. При проходе каждым набором весов формируется карта признаков. Поскольку появляется много независимых карт признаков на одном слое, то сеть становится многоканальной.

В каждом слое свёртки для каждого канала свой фильтр, ядро свёртки которого обрабатывает предыдущий слой по фрагментам. Результат применения различных фильтров объединяется. Так получаются слои объединения. Операция субдискретизации выполняет уменьшение размерности сформированных картпризнаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности. За счёт данной операции, помимо ускорения дальнейших вычислений, сеть становится инвариантной к масштабу входного изображения.

После начального слоя сигнал проходит серию свёрточных слоёв, в которых чередуется операции свёртки и объединения(pooling). Чередование слоёв позволяет составлять карты признаков: на каждом следующем слое карта уменьшается в размере, а количество каналов увеличивается.

Практически это означает способность распознавания сложных иерархий признаков.

После прохождения нескольких слоев карта признаков вырождается в вектор или скаляр, но таких карт признаков становится сотни. На выходе свёрточных слоёв сети дополнительно устанавливают несколько слоев полносвязной нейронной сети (например, перцептрон), на вход которому подаются конечные карты признаков.

Гиперпараметры сети

Гиперапараметрами свёрточной нейронной сети являются:

Узкая и широкая свёртки (wide and narrow convolutions): дополнение нулями (zero padding) позволяет сделать свёртку широкой в случае, когда, например, первый элемент матрицы не имеет соседних элементов слева и сверху. Без использования дополнения нулями получаем узкую свёртку;

Размер шага (stride): Размер шага определяет величину сдвига фильтра на каждом шаге. Чем больше шаг, тем меньше фильтр применяется и тем меньше размер выходной матрицы. Обычно использует шаг равный единице, однако больший шаг может позволить построить модель, поведение которой будет напоминать рекурсивную нейронную сеть (т.е. свёрточная сеть с большим шагом будет выглядеть как дерево);

Слои объединения: Слои объединения помогают сократить размерность выходной информации, при этом сохраняя самую заметную информацию. Например, если фильтр определяет, содержит ли предложение отрицание ("not good"). Если где-то в предложении есть эта фраза, то результат применения фильтра к этому региону даст большое значение, но малое для других регионов. После применения операции максимума для региона, остается только информация, появлялось ли заявленное отрицание в предложение, однако информация о том, где оно появлялось, исчезает. То есть информация о местоположении пропадает, а локальная информация остаётся (очевидно, что “not good” сильно отличается от “good not”);

Каналы (channels): каналы — это разные “взгляды” на входные данные. Например, в распознавании изображений, у нас обычно три канала — RGB. В обработке естественного языка такими каналами могут являться различные векторные представления слов (word2vec или GloVe), предложение на разных языках или перефразированные предложения.

Типовая структура

Структура сети является однонаправленной, а для обучения обычно используется метод обратного распространения ошибки. Сеть состоит из большого количества слоёв (см. рис. 1.7).

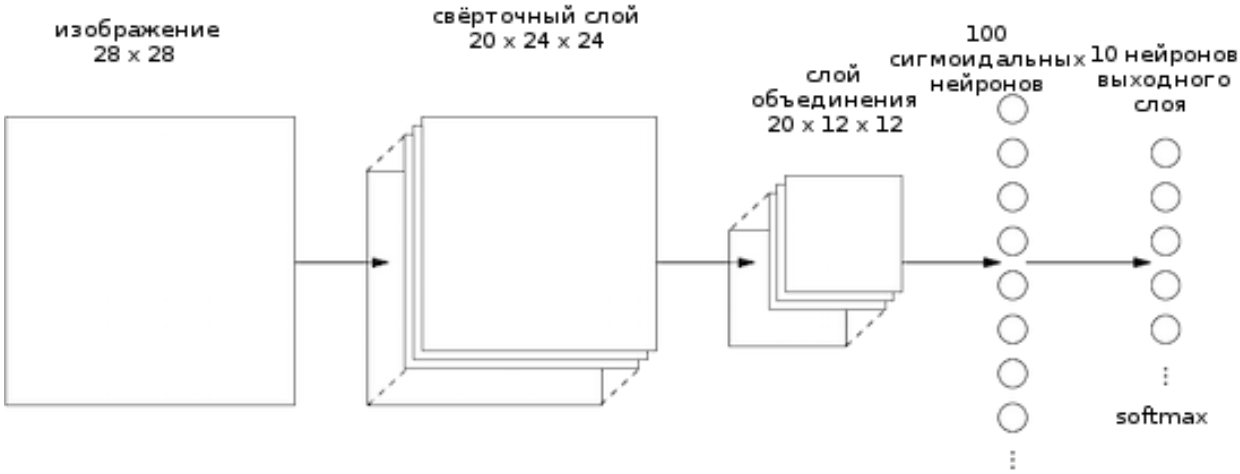

Рисунок 1.7 — Пример архитектуры свёрточной нейронной сети для рас-познавания объекта на изображении размерности 28x28 px

1.3. Применение свёрточных нейронных сетей в анализе тональности текста

На вход нейронной сети будет подаваться матрица, количество строк которой зависит от размерности словаря, а ширина фильтров равна количеству столбцов этой матрицы (то есть используемой размерности для кодирования каждого слова). Высота (или размер фрагмента входных данных) может меняться, но обычно она составляет около 2—5 слов. Первые слои представляют слова в виде низкоразмерных векторов. Следующий слой выполняет свёртки над векторными представлениями слов, используя фильтры разных размеров (то есть они захватывают 3-5 слов одновременно). Затем производится пулинг (max-pool) над результатом свёртки. К полученному длинному вектору признаков добавляем регуляризацию (dropout в этом случае). Наконец, происходит классификация результата с помощью слоя softmax (см. рис. 1.8).

В качестве входов задаются не только стандартные X и Y, но и вероятность того, что нейрон окажется в слое дропаута (дропаут задаётся только во время тренировки сети). Первый слой — слой представления слов в виде векторов word2vec — является таблицей преобразования (соответствия). Результат применения этого слоя — трёхмерный тензор размерности [None,sequence_length, embedding_size].

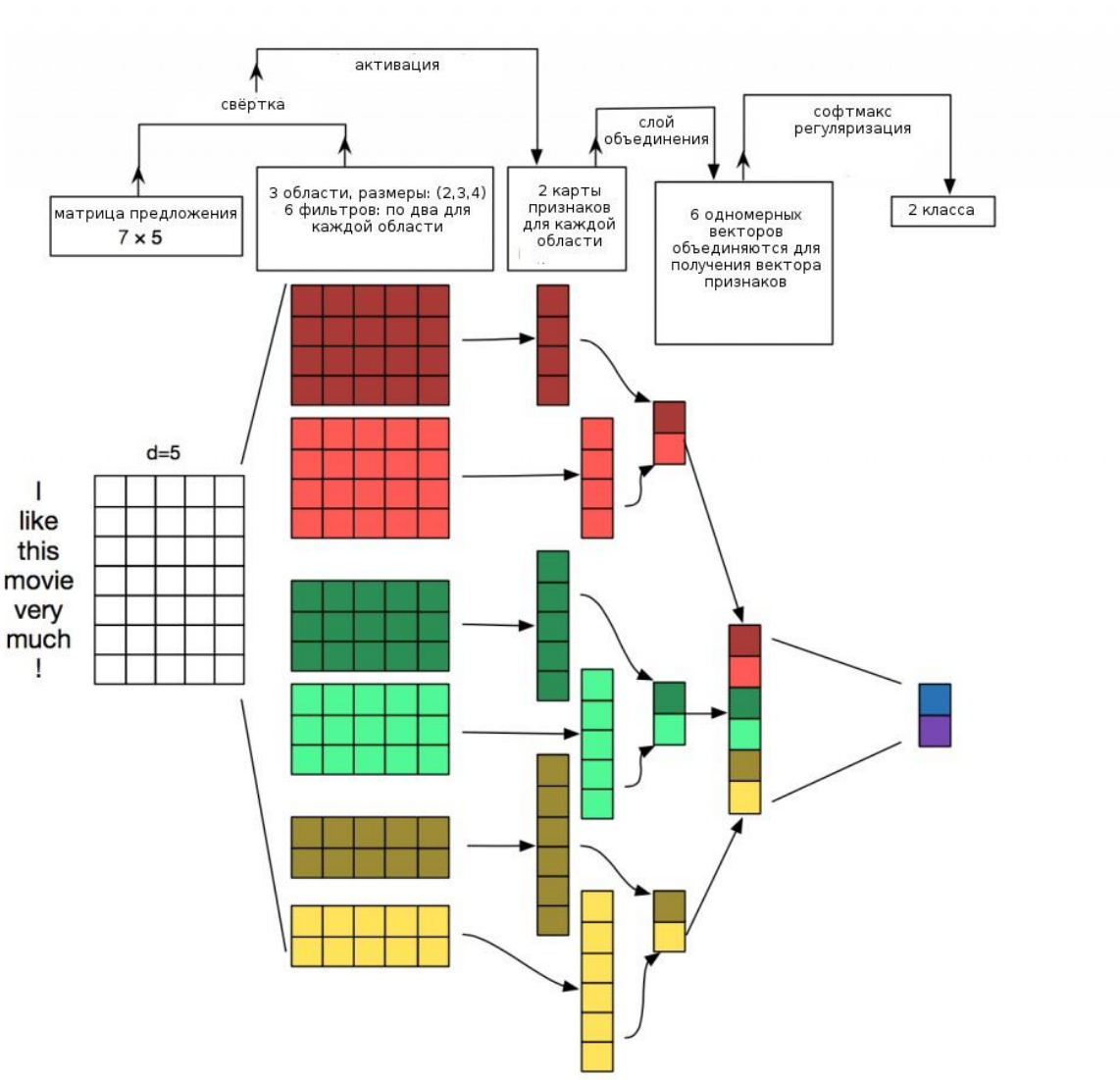

Рисунок 1.8 — Пример архитектуры свёрточной нейронной сети для классификации предложений

Следующий слой свёртки принимает на вход 4-мерный тензор (батч, ширина, высота и канал). К полученному на предыдущем шаге тензору просто

прибавляется новое измерение [None, sequence_length, embedding_size,1]. Каждый фильтр проходит полностью через всё представление, отличие только в том, сколько он обрабатывает слов. В данной работе используется узкая свёртка (narrow convolution) — поэтому на выходе тензор имеет форму [1, sequence_length - filter_size + 1, 1, 1]. После применения пулинга над выходом определенного фильтра получится тензор формы [batch_size, 1, 1, num_filters] По существу это и есть вектор признаков, последнее измерение которого соответствует нужным нам признакам. После получения всех тензоров, нужно объединить их в один длинный вектор признаков формы [batch_size, num_filters_total]

Самым популярным методом регуляризации свёрточных нейронных сетей является dropout. Dropout блокирует группу нейронов случайным образом. Это заставляет их "изучать"полезные признаки самостоятельно. Группа нейронов, которая принимает участие в обучении, определяется dropout_keep_prob входом в сети.

После получения обработанного вектора признаков, можно делать предсказания к какому классу относится данная рецензия. Необходимо перемножить матрицы и выбрать класс с наибольшим результатом. Слой softmax поможет нормализовать полученные вероятности.

Используя полученные значения, становится возможным определить функцию потерь. Наша цель — минимизировать потери, которые как раз измеряют ошибку (погрешность) нейронной сети. Для этого используется перекрёстная энтропия.

В работе используется функция softmax_cross_entropy_with_logits.

Затем берётся среднее значение потерь. Как классифицируются предложения с помощью данной архитектуры:

Определить размерности трёх фрагментов данных: 2, 3 и 4, для каждого фрагмента существует два фильтра;

Каждый фильтр выполняет свёртку над матрицей предложений и генерирует карты признаков (feature maps);

Слой 1-max pooling обрабатывает каждую карту, то есть сохраняется наибольшее число из каждой карты. Полученный одномерный вектор признаков из карт объединяется в вектор признаков для предпоследнего слоя;

Последний (softmax) слой получает этот вектор на вход и использует его для классификации предложения (бинарная классификация).

Благодаря использованию свёрточных слоев, количество параметров в них резко сокращается, и обучить сеть становится гораздо проще. А используя различные виды регуляризации (особенно dropout), получилось значительно уменьшить переобучение сети. Наконец, применение ReLU вместо сигмоидальных нейронов помогло ускорить обучение.

Может показаться, что идея свёрточных нейронных сетей не очень применима для задач естественного языка: действительно, если на изображении соседние пиксели в основном являются частью одного и того же объекта, то это неверно в случае слов в предложении (части фраз могут быть разделены другими словами). Возможно, рекуррентные нейронные сети — более интуитивно понятная модель (предложение представлено в виде дерева разбора), однако это не значит, что свёрточные сети совсем не применимы для поставленных в работе задач.

Рекуррентные нейронные сети

В сетях прямого распространения используется единственный вход, который полностью определяет активации всех нейронов в оставшихся слоях. Такую сеть невозможно обучить предсказывать события, например, в сюжете фильма — неясно, как бы могла быть использована информация о предыдущих событиях в фильме. Рекуррентные нейронные сети призваны решить эту проблему. Имея внутри циклы, RNN позволяет информации сохраняться: поведение скрытых нейронов будет определяться не только

активацией в других скрытых слоях, но и полученными ранее активациями самих нейронов.

RNN может быть представлена в качестве множества копий одной и той же нейронной сети, где каждая копия передает сообщение следующей копии. То есть, имея цепеобразную структуру, как последовательности или списки, RNN является естественной архитектурой нейронной сети, используемой для таких данных.

RNN способны обусловливать модель по всем предыдущим обработанным словам из корпуса текстов. На рис. 3.10 прямоугольник является скрытым слоем на временном шаге t. Каждый слой содержит нейроны (см. рис. 3.9), каждый из которых выполняет операцию линейной матрицы на своих входах, за которой следует нелинейная операция (например, tanh). На каждом временном шаге выходные данные предыдущего шага вместе со следующим вектором слова xt текста, представляет собой входные данные для скрытого слоя для создания предсказания yˆt и признаков ht

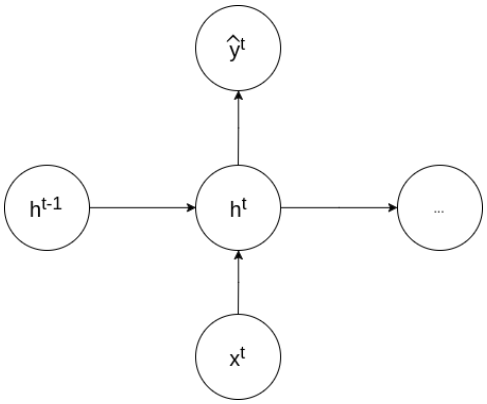

Рисунок 1.9 — Нейрон рекуррентной сети

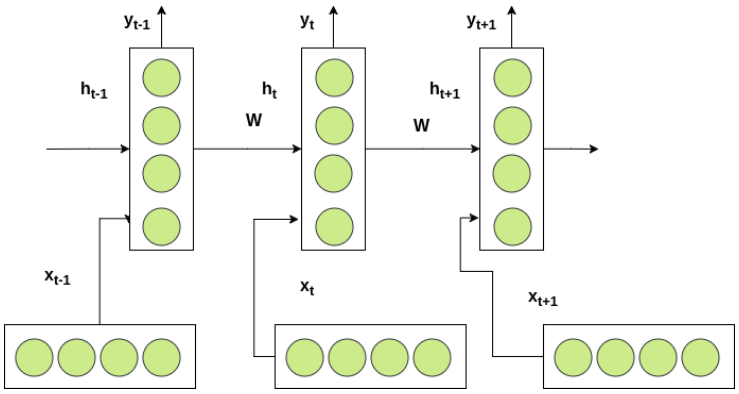

Рисунок 1.10 — Рекуррентная нейронная сеть, три временных шага

Объем памяти, необходимый для запуска слоя RNN, пропорционален количеству слов в корпусе текстов. То есть предложение, состоящее из k слов, будет храниться в памяти как k векторов. Размер матрицы весов W не масштабируется в соответствии с размером корпуса текстов. Для рекуррентной сети, состоящей из 1000 рекуррентных слоёв, размер матрицы всегда будет 1000 × 1000, в независимости от размера корпуса текстов.

Рекурсивная и рекуррентная нейронные сети

Рекуррентные нейронные сети повторяются (recurring) с течением времени. Пусть необходимо предсказать следующий символ после “hell” и “угадывающего” следующую букву слова — “o”

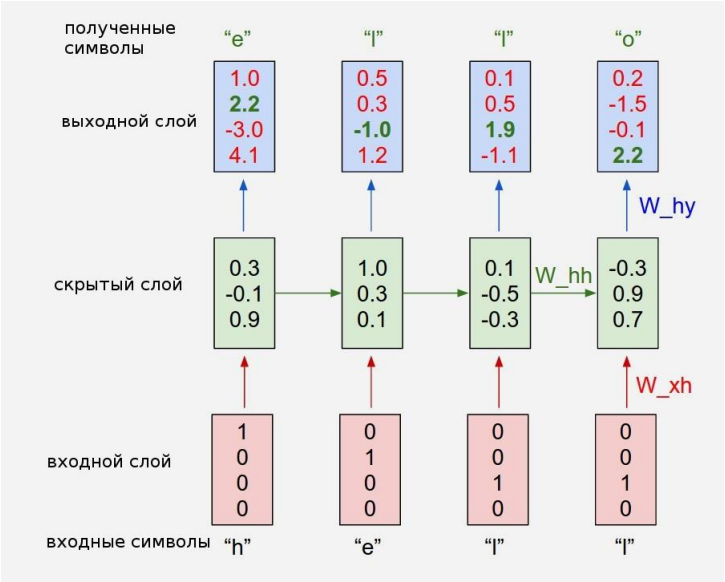

Рисунок 1.11 — Схема нейрона, принимающего на вход последовательность

Важно различать понятия рекуррентной и рекурсивной нейронной сети. Рекурсивная нейронная сеть – это обобщение рекуррентной. В рекуррентной сети веса общие (и размерность остаётся одинаковой) по всей длине последовательности. В рекурсивной сети веса также общие в каждом узле. Это значит, то все W_xh веса будут одинаковыми (общими) и таким же будет вес W_hh из-за того, что всё происходит в единственном нейроне, развёртывающимся во времени (см. рис. 3.12).

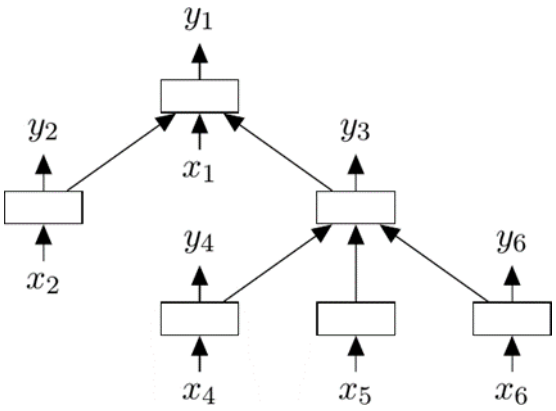

Рисунок 1.12 — Схема рекурсивной нейронной сети

Если сети используются для генерации новых символов, то подойдут рекурсивные сети. Однако для генерации дерева разбора лучше использовать рекуррентные сети.

Проблема исчезающего градиента на примере языковой модели

Пусть имеется языковая модель, с помощью которой необходимо предсказать следующее слово, используя предыдущие. Когда разрыв между важной информацией и отрывком, где она необходима, невелик, RNN можно обучить использовать информацию, полученную ранее (“Облака в небе”). Но если необходим контекст, как в примере “Я вырос во Франции. Я свободно говорю по-французски”, то разрыв между существенной информацией и местом вставки становится шире.

К сожалению, по мере роста разрыва, RNN невозможно обучить связывать информацию. Одной из проблем RNN является то, что её ранние модели очень сложно обучать из-за неустойчивого градиентного спуска.

Обучение в предыдущих слоях происходит очень медленно, так как градиент становится всё меньше и меньше при обратном распространении. То есть если сеть работает довольно долго, то градиент может стать крайне неустойчивым.

1.4.Обзор средств разработки

Фреймворк TensorFlow

TensorFlow - это библиотека программного обеспечения с открытым исходным кодом для задач машинного обучения, разработанная Google. Она позволяет создавать и обучать нейронные сети различной архитектуры для обнаружения и распознавания образцов и поиска взаимосвязей. TensorFlow также включает в себя TensorBoard, который представляет собой средство визуализации в браузере для оценки эффективности обучения и сетевых параметров модели.

TensorFlow достигает своей производительности благодаря распараллеливанию задач между центральным и графическими процессорами. Ядро каждой операции реализовано на C++ с применением библиотек Eigen и cuDNN для лучшей производительности.

Каждое вычисление в TensorFlow представляется как граф потока данных, он же граф вычислений. Граф вычислений является моделью, описывающей как, будут выполняться вычисления. Важно заметить, что составление графа вычислений и выполнение операций в заданной структуре — два разных процесса. Граф состоит из плэйсхолдеров (tf.Placeholder), переменных (tf.Variable) и операций. В нём производится вычисление тензоров — многомерных массивов, которые, впрочем, могут быть числом или вектором.

Графы выполняются в сессиях (tf.Session). Существуют два типа сессий — обычные и интерактивные (tf.InteractiveSession); интерактивная сессия подходит для выполнения в консоли. Сессия хранит состояние переменных (Variables) и очередей (queues). Явное создание сессий и графов гарантирует надлежащее освобождение ресурсов памяти

В графе каждая вершина имеет 0 или больше входов и 0 или больше выходов, и представляет собой реализацию операции. Тензоры представляют собой рёбра графа, а именно массивы произвольного размера (тип массива указывается во время построения графа). Особые вершины, управляющие зависимости (control dependencies), также могут быть в графе: они указывают, что исходный узел для контрольной зависимости должен закончить выполнение до того, как узел получателя контрольной зависимости начнет выполняться.

Каждая операция имеет название и представляет собой абстрактное вычисление (например, суммирование). У операции могут быть аттрибуты: например, возможность сделать операцию полиморфной для разных типов

тензоров. Ядро — специфическая реализация операции, которая может выполнена на определенном типе устройства (центральный или графический процессор).

Переменная — особый вид операции, возвращающий указатель на постоянно меняющийся тензор: такая переменная не исчезает после единичного использования графа. Указатели на подобные тензоры передаются многочисленным операциям, которые затем изменяют указанный тензор.

В задачах машинного обучения, параметры модели обычно хранят тензоры в переменных, которые обновляются на каждом шаге обучения. Данная работа выполнена на единственном устройстве с применением CPU.

Особенности

В Tensorflow существуют несколько форм параллелизма:

Параллелизм в отдельных операциях (например, tf.nn.conv2d () и tf.matmul ()). Эти операции имеют эффективные параллельные реализации для многоядерных процессоров и графических процессоров, и TensorFlow использует эти реализации во всех возможных случаях;

Параллелизм между операциями. TensorFlow использует представление графа вычислений и там, где есть два узла, которые не связаны прямым путем, они могут выполняться параллельно;

Параллелизм между копиями моделей. Стандартная схема для параллельного обучения — разделить данные между workers, провести одинаковые вычисления для разных данных и обменяться параметрами модели между workers.

Уникальность Tensorflow заключается в возможности проводить частичные подграфовые вычисления. Эта особенность позволяет сделать разбиение нейронной сети, а значит можно использовать распределенное обучение. Также:

- Tensorboard: визуализация модели и возможность исследовать порядок вычислений в графе;
- Tensorflow может использована как на мобильных, так и на более мощных устройствах.

В Tensorflow производные задаются автоматически: этот процесс называется автоматическим дифференцированием.

Традиционные методы оценки производной сложно реализовать на практике, так как они имеют ряд недостатков. Например, применение метода конечных разностей требует обоснованного выбора значения приращения аргумента. Однако существует способ автоматического вычисления вместе с функцией $f(x_0)$ её производной $f'(x_0)$ при некотором значении аргумента $x = x_0$. Данный метод называется автоматическим дифференцированием, так как вычисления значение f(x0) и f'(x0) осуществляется одновременно на основе исходного кода только функции f(x). Он позволяет получить точное (до ошибок округления) значения производной, а программу вычислений достаточно выполнить только один раз.

Tensorflow использует обратный режим автоматического дифференцирования для операций градиентов и метода конечных разностей для тестов, которые проверяют правильность работы градиента.

Обычно в системах автоматического дифференцифирования оператор (сумма, разность) определён вместе с его производными. То есть после написания функции, в которой определено несколько операторов, программа может сама выяснить, как вычислить соответствующие производные (используя граф вычислений и цепное правило). Выгода очевидна, так как не нужно самостоятельно разрабатывать математические операции и численно проверять каждую производную.

```
tf.reset_default_graph()
x = tf.Variable(0.)
y = tf.square(x)
z = tf.gradients([y], [x])
```

Рисунок 2.1 — Фрагмент кода для демонстрации автоматического дифференцирования

Pandas

Pandas - это библиотека с открытым исходным кодом, предоставляющая высокопроизводительные, простые в использовании структуры данных и инструменты анализа для языка программирования Python.

Python давно отлично подходит для сбора данных, но в меньшей степени для анализа и моделирования данных. pandas помогает заполнить этот пробел, позволяя вам выполнять весь рабочий процесс анализа данных на Python без необходимости перехода на более подходящий для анализа данных язык, например R.

Особенности:

- Быстрый и эффективный объект DataFrame для обработки данных с интегрированной индексацией;

- Инструменты для чтения и записи данных между структурами данных в памяти и различными форматами: CSV, Microsoft Excel, базы данных SQL и быстрый формат HDF5;

- Интеллектуальное выравнивание данных и интегральная обработка отсутствующих данных: автоматическое выравнивание на основе меток в вычислениях и легкое манипулирование беспорядочными данными в упорядоченной форме;

- Интеллектуальная нарезка на основе ярлыков, удобная индексация и подмножество больших наборов данных;

- Преобразование данных с помощью мощной группы посредством механизма, позволяющего выполнять операции split-apply-comb на наборах данных;

- Высокопроизводительное объединение наборов данных;

- Pandas используется в самых разных академических и коммерческих областях, включая финансы, неврологию, экономику, статистику, рекламу, веб-аналитику и многое другое.

Scikit-learn

Scikit-learn - это библиотека для машинного обучения для языка программирования Python. Она имеет различные алгоритмы классификации, регрессии и кластеризации, случайные леса, повышение градиента, k-среднее и DBSCAN, предназначен для взаимодействия с численными библиотеками NumPy и SciPy.

Возможности:

- Кластеризация (Clustering): для группировки неразмеченных данных, например, метод k-средних (k-means)

- Перекрестная проверка (Cross Validation): для оценки эффективности работы модели на независимых данных

- Наборы данных (Datasets): для тестовых наборов данных и для генерации наборов данных с определенными свойствами для исследования поведенческих свойств модели

- Сокращение размерности (Dimensionality Reduction): для уменьшения количества атрибутов для визуализации и отбора признаков (Feature Selection), например, метод главных компонент (Principal Component Analysis)

- Алгоритмические композиции (Ensemble Methods): для комбинирования предсказаний нескольких моделей

- Извлечение признаков (Feature Extraction): определение атрибутов в изображениях и текстовых данных

- Отбор признаков (Feature Selection): для выявления значимых атрибутов на основе которых будет построена модель

- Оптимизация параметров алгоритма (Parameter Tuning): для получения максимально эффективной отдачи от модели

- Множественное обучение (Manifold Learning): для нелинейного сокращения размерности данных

- Алгоритмы обучения с учителем (Supervised Models): огромный набор методов не ограничивается обобщенными линейными моделями (Generalized Linear Models), дискриминантным анализом (Discriminate Analysis), наивным байесовским классификатором (Naive Bayes), нейронными сетями (Neural Networks), методом опорных векторов (Support Vector Machines) и деревьями принятия решений (Decision Trees).

1.5.Классические методы классификации

Процесс предобработки данных (каждой рецензии) состоит из следующих шагов:

• Удалить разметку HTML;

• Удалить все символы, кроме букв и пробелов;

• Из полученного набора слов удалить стоп-слова.

Следующей задачей является преобразование каждой рецензии в векторное представление. Для оценки эффективности каждого из методов будут использованы две модели векторного представления слов: мешок слов и Word2Vec.

Логистическая регрессия

Статистическая модель, используемая для предсказания вероятности возникновения некоторого события.

Scikit-learn (sklearn.linear_model.LogisticRegression). В работе добавлен единственный параметр: random_state = 1.

Наивный байесовский классификатор

Простой вероятностный классификатор, основанный на применении Теоремы Байеса со строгими предположениями о независимости элементов вектора признаков. Достоинством наивного байесовского классификатора является малое количество данных для обучения, необходимых для оценки параметров, требуемых для классификации.

Scikit-learn (sklearn.naive_bayes.GaussianNB). В работе используем наивный байесовский классификатор Гаусса — распределение вероятностей признаков совпадает с функцией Гаусса (то есть нормальное распределение). σy и μy рассчитаны по методу максимального правдоподобия.

$$P(x_i \mid y)=1/\sqrt{(2\pi\sigma_y^2)} \exp(-(x_i-\mu_y)^2/(2\sigma_y^2)) \quad (2.1.1)$$

Случайный лес (random forest)

Алгоритм заключается в использовании комитета решающих деревьев. Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев. Оптимальное число деревьев подбирается таким образом, чтобы минимизировать ошибку классификатора на тестовой выборке. Использует усреднение для повышения точности прогнозирования и контроля избыточной подгонки. Размер подвыборки всегда совпадает с размером оригинальной выборки.

Scikit-learn (sklearn.ensemble.RandomForestClassifier). Количество деревьев (n_estimators) в работе равно 100. Чем их больше, тем лучше, но может возникнуть как проблема переобучения, так и проблемы с памятью устройства.

Метод опорных векторов (SVM)

Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Стандартная функция из scikitlearn (sklearn.svm.SVC) не подходит, поскольку временная сложность данного алгоритма квадратична. Эффективность метода опорных векторов значительно снижается, если количество признаковых описаний очень велико. Он имеет большую гибкость в выборе штрафов и функций потерь и должен лучше масштабироваться для большого количества образцов. Оптимизация функции потерь SVM с помощью градиентного спуска:

$$L(\omega,D)=1/2\ \|\omega\|_2^2+C\sum_{i=1}^{N}\ [\max(0,1-y_i\ (\omega x_i+b)),] \quad (2.1.2)$$

где

$$D=\{(x_i,y_i)\}_{i=1}^{N},x_i \in Rd \text{ и } y_i \in\{-1,+1\} \quad (2.1.3)$$

Оптимизация функции потерь — (regularization_loss + hinge_loss). Для случая word2vec будети спользован SGDClassifier из sklearn.linear_model с ошибкой l2.

Свёрточная нейронная сеть

Векторные представления данных осуществлены с помощью модели Word2Vec (Skip-gram Model).

Параметры сети

Параметрами свёрточной нейронной сети являются:

Размерность векторного представления слова = 150;

Размер батча = 50;

Темп обучения = 0.001;

Количество шагов обучения = 2950;

Dropout = 0.8;

Размерность фильтров = (2, 3, 4);

L2 regularization = 4.

В качестве функции оптимизации используется Adam (Adaptive Moment Estimation). Его отличительными особенностями являются:

• оценка первого момента вычисляется как скользящее среднее;

• так как оценки первого и второго моментов инициализируются нулями, используется небольшая коррекция, чтобы результирующие оценки не были смещены к нулю.

Метод также инвариантен к масштабированию градиентов.

Рекуррентная нейронная сеть с LSTM-блоками Векторные представления данных осуществлены с помощью модели Word2Vec (Skip-gram Model). LSTM считаются наилучшей архитектурой для анализа тональности текста. Нейронные сети, составленные из LSTM-модулей, особенно хорошо обрабатывают отрицание (negation), если в ячейке есть

projection unit (то есть больше памяти для сети).Архитектура реализованной сети:

Слой векторного представления слов: преобразует каждый вход (тензор из k слов) в тензор k N-мерных векторных представлений слов (N— размер представления). Каждому слову соответствует вектор весов, который необходимо изучить во время процесса обучения сети:

– Из каждой рецензии удаляются все символы, кроме пробелов и букв; текст представлен только строчными буквами;

– Создаётся словарь, из каждой рецензии удаляются самые редко встречающиеся слова (которые скорее всего появились в результате грамматической ошибки);

– Создаётся тензор, представляющий каждую рецензию;

– Каждый тензор дополняется нулями соответственно рецензии, имеющей максимальную длину.

• RNN слой: создан из LSTM-блоков с обёрткой dropout'a. Веса LSTM изучаются во время обучения. RNN слой развёртывается динамично, беря на вход k векторных представлений и выдаёт M-мерные вектора, где M — количество LSTM-модулей в блоке;

• Слой softmax: выход RNN слоя усредняется через k временных шагов. Выдаёт тензор размерности M, который используется для вычисления вероятностей для задачи классификации.

Параметры сети

Параметрами рекуррентной нейронной сети являются:

• Размер слоя векторного представления = 50;

• Размер батча = 100;

• Темп обучения = 0.1;

• Количество шагов обучения = 1000;

• Dropout = 0.5;

• Скрытый размер LSTM слоя (количество LSTM-модулей в блоке) = 50.

1.6.Обзор аналогов

Компьютеры могут выполнять автоматический анализ цифровых текстов, используя элементы машинного обучения, такие как скрытый семантический анализ, метод опорных векторов, «мешок слов» и семантическая направленность в этой области. Более сложные методы пытаются определить обладателя настроений (то есть человека) и цель (то есть сущность, в отношении которой выражаются чувства). Чтобы определить мнение с учётом контекста, используют грамматические отношения между словами.

Отношения грамматической связанности получают на основе глубокого структурного разбора текста. Анализ тональности может быть разделен на две отдельные категории:

- ручной (или анализ тональности экспертами);
- автоматизированный анализ тональности.

Наиболее заметные различия между ними лежат в эффективности системы и точности анализа. В компьютерных программах автоматизированного анализа тональности применяют алгоритмы машинного обучения, инструменты статистики и обработки естественного языка, что позволяет обрабатывать большие массивы текста, включая веб-страницы, онлайн-новости, тексты дискуссионных групп в сети Интернет, онлайн-обзоры, веб-блоги и социальные медиа.

Методы, основанные на правилах и словарях

Этот метод основан на поиске эмотивной лексики (лексической тональности) в тексте по заранее составленным тональным словарям и правилам с применением лингвистического анализа. По совокупности найденной эмотивной лексики текст может быть оценен по шкале,

содержащей количество негативной и позитивной лексики. Данный метод может использовать как списки правил, подставляемые в регулярные выражения, так и специальные правила соединения тональной лексики внутри предложения. Чтобы проанализировать текст, можно воспользоваться следующим алгоритмом: сначала каждому слову в тексте присвоить его значение тональности из словаря (если оно присутствует в словаре), а затем вычислить общую тональность всего текста путём суммирования значения тональностей каждого отдельного предложения.

Основной проблемой методов, основанных на словарях и правилах, считается трудоёмкость процесса составления словаря. Для того, чтобы получить метод, классифицирующий документ с высокой достоверностью, термины словаря должны иметь вес, адекватный предметной области документа. Например, слово «огромный» по отношению к объему памяти жёсткого диска является положительной характеристикой, но отрицательной по отношению к размеру мобильного телефона. Поэтому данный метод требует значительных трудозатрат, так как для хорошей работы системы необходимо составить большое количество правил. Существует ряд подходов, позволяющих автоматизировать составление словарей для конкретной предметной области (например, тематика ресторанов или тематика мобильных телефонов).

Машинное обучение с учителем

В наше время наиболее часто используемыми в исследованиях методами являются методы на основе машинного обучения с учителем. Сутью таких методов является то, что на первом этапе обучается машинный классификатор (например, байесовский) на заранее размеченных текстах, а затем используют полученную модель при анализе новых документов. Опишем краткий алгоритм:

вначале собирается коллекция документов, на основе которой обучается машинный классификатор;

каждый документ раскладывается в виде вектора признаков(аспектов), по которым он будет исследоваться;

указывается правильный тип тональности для каждого документа;

производится выбор алгоритма классификации и метод для обучения классификатора;

полученную модель используем для определения тональности документов новой коллекции.

Машинное обучение без учителя

В основе этого подхода лежит идея, что термины, которые чаще встречаются в этом тексте и в то же время присутствуют в небольшом количестве текстов во всей коллекции, имеют наибольший вес в тексте. Выделив данные термины, а затем определив их тональность, можно сделать вывод о тональности всего текста.

Метод, основанный на теоретико-графовых моделях

В основе этого метода используется предположение о том, что не все слова в текстовом корпусе документа равнозначны. Какие-то слова имеют больший вес и сильнее влияют на тональность текста. При использовании этого метода анализ тональности разбивается на несколько этапов:

- построение графа на основе исследуемого текста;
- ранжирование его вершин;
- классификация найденных слов;
- вычисление результата.

Для классификации слов используется тональный словарь, в котором каждому слову соотносится оценка, например, «положительная», «отрицательная» или «нейтральная». Для получения конечного результата

нужно вычислить значения двух оценок: положительной составляющей текста и отрицательной. Для того, чтобы найти положительную составляющую текста необходимо найти сумму тональностей всех положительных терминов текста с учетом их веса. Значение отрицательной составляющей текста находится аналогичным образом. Для итоговой оценки тональности всего текста нужно вычислить отношение этих составляющих по формуле: T=P/N, где T — итоговая оценка тональности, P — оценка положительной составляющей текста и N — негативная составляющая текста

В настоящее время задача построения классификатора для определения тональности текста решается с помощью нейросетевых моделей, так как эффективность архитектур, использованных в упомянутых работах, значительно выше, чем у классических линейных алгоритмов.

Существует множество библиотек для реализаций алгоритмов машинного обучения. Существуют два типа фреймворков: символьные и императивные. В символьных фреймворках гораздо больше возможностей исполь-зовать память многократно, а оптимизация на основе графов зависимостей осуществляется автоматически. Самыми популярными символьными (symbolic) фреймворками в настоящее время являются TensorFlow и Theano.

В отличие от Theano, Tensorflow не ориентирован только на обучение нейронных сетей, поэтому можно использовать коллекции графов и очереди в качестве составных частей для высокоуровневых компонентов.

Если необходимо обучать масштабные модели и использовать много внешней памяти, то Theano будет очень медленно работать из-за необходимости компиляции кода C/CUDA в бинарный код.

TensorFlow имеет прозрачную модульную архитектуру с множеством фронт-эндов. В архитектуре Theano разобраться довольно непросто: весь код - это Python, где код C/CUDA упакован как строка Python. В таком коде сложно

ориентироваться, его непросто отлаживать и проводить рефакто-ринг. Более того, визуализация графов в TensorFlow реализована значительно эффективнее, чем в Theano [1].

Векторные представления слов для линейных алгоритмов будут представлены двумя моделями: Word2Vec и мешок слов (bag of words), а для классификаторов на основе нейронных сетей будет использована только модель Word2Vec. С помощью инструмента для построения векторных моделей Gensim будет обучена модель Word2Vec. С применением TensorFlow бу-дут реализованы свёрточная нейронная сеть и рекуррентная нейронная сеть с LSTM-блоками.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И ТЕОРИТИЧЕСКОЕ ОБОСНОВАНИЕ СИСТЕМЫ.

2.1.Постановка задачи

Целью данной работы является разработка алгоритмов для анализа тональности текста на основе глубоких нейронных сетей (свёрточной и рекуррентной), а также сравнение их эффективности с другими классификаторами. В качестве опытного образца были использованы рецензии веб-сайта RottenTomatoes [1] — набор из 5331 позитивных и 5331 негативных рецензий. Для разработки использовались библиотеки Pandas, Scikit-Learn и PyMorphy2, а также фреймворк TensorFlow как средство анализа.

Функциональные требования к системе

1. Предоставление пользователю возможности проанализировать собственный текст.
2. Реализация алгоритма обновления данных в реальном времени.
3. Распознать тональность отзыва.
4. Разбить отзыв на отдельные слова.
5. Удаление лишней информации из текста (нежелательные символы).
6. Провести морфологический анализ текста.
7. Провести лемматизацию.
8. Построить векторную модель из текста.
9. Обучить нейронную сеть.

Не функциональные требования

1. Наличие персонального компьютера, имеющего доступ к интернету.
2. Обеспечить запись в базу данных оценки анализируемого текста.
3. В случае невозможности правильно анализировать текст, выдавать сообщение об ошибке.
4. Переносимость на различные компьютерные платформы.
5. Простота в эксплуатации опытным оператором.

Программные требования

- Сервер с поддержкой python 3 и баз данных на языке SQL
- Сетевые протоколы

Операционные системы, поддерживаемые SQL, содержат встроенное ПО с поддержкой сетевых протоколов: именованные каналы, общая память и TCP/IP.

- Жесткий диск

Требуется минимум 6 Гб свободного места на диске.

- Монитор

Требуется монитор с разрешением 800x600 пикселей или более высоким.

- Интернет

Требуется доступ в Интернет для поддержки функциональных средств Интернета.

Требования к оборудованию

- ОЗУ Минимально требуется: 1 Гб

Рекомендуемые требования: 4 Гб с увеличение по мере роста размеров базы данных.

- Производительность процессора

Минимально требуется: процессор с архитектурой x64 и тактовой частотой 1.4 ГГц.

Рекомендуемые требования: процессор с архитектурой x64 и тактовой частотой 2.0 ГГц или выше.

- Тип процессора

Процессор архитектуры x64: AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой Intel EM64T.

Алгоритм обучения нейронной сети

Для того чтобы нейронная сеть начала выполнять свои задачи ее необходимо обучить, процесс обучения происходит по принципу показанном на рис. 2.4

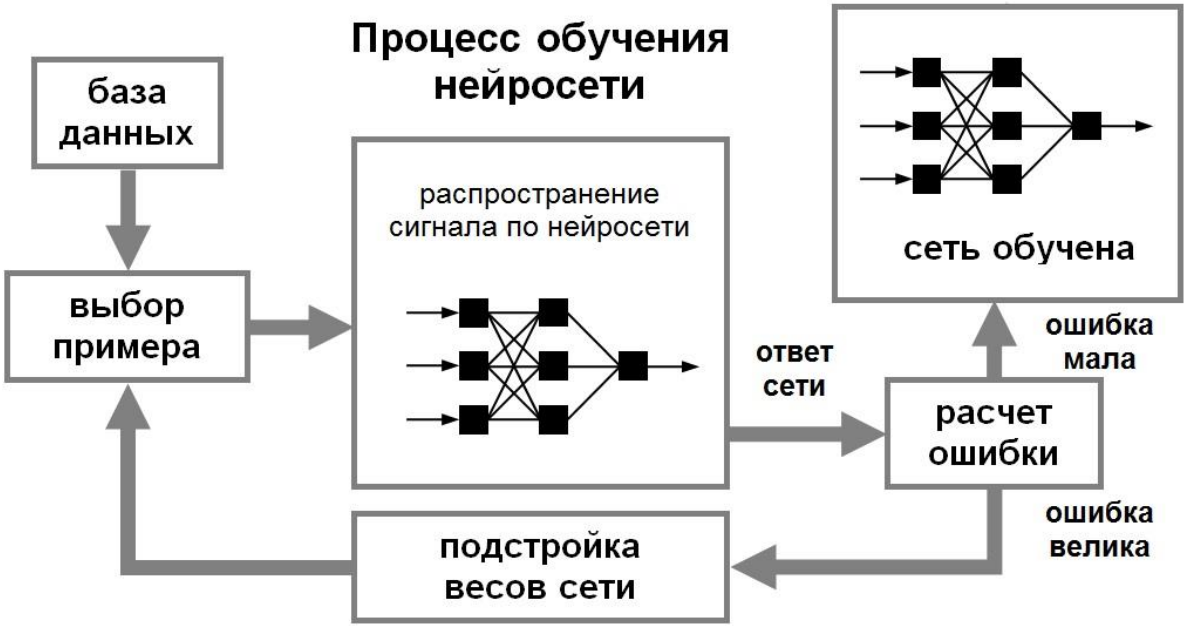

Рисунок 2.4 — Доля корректных прогнозов (accuracy): синий график – обучение, красный — проверка

2.2.Алгоритм анализа данных в приложении

1. Получаем данные из другого источника (базы данных, пользовательского интерфейса)

2. Удаляем лишнюю информацию из предлагаемого текста оставляя только русские буквы

3. Производим морфологический анализ текста, и лематизируем текст

4. Строим модель:

- Схема n-грамм: (1, 3) (униграммы + биграммы + триграммы);

- Метод векторизации: Word2Vec;

- Тип модели: Рекуррентная нейронная сеть с LSTM-блоками;

Параметры модели: penalty – l2, alpha – 0.000001, loss – log.

5.Обучаем нейронную сеть по полученным данным

2.3.Показатели качества нейронной сети

Доля корректных прогнозов (accuracy) — процент ошибок, допускаемых классификатором.

Следующие показатели будут использованы только для классических методов классификации:

• Мера точности (precision) — отношение tp к (tp+fp), где tp — количество истинных положительных величин, а fp — количество ложных положительных величин. То есть мера точности характеризует сколько полученных от классификатора позитивных решений считаются верными;

• Мера полноты (recall) — отношение tp к (tp + fn), где fn — количество ложных отрицательных величин. Мера полноты устанавливает умение классификатора узнавать равно как возможно наибольшее количество позитивных решений с прогнозируемых;

• Мера F1 — среднее гармоническое меры точности и меры полноты. Определяет лиминальное свойство классификатора;

• Носитель меры (support) — число информации любого с классов. Наиболее жесткое определение: минимальное закрытое большое число, в котором сконцентрирована степень.

Метод мешка слов (см. табл. 2.2.1, 2.2.2, 2.2.3 и 2.2.4).

Таблица 2.2.1 — Логистическая регрессия

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.75	0.76	0.75	1192
1	0.74	0.74	0.74	1139
total	0.74	0.74	0.74	2331
Достоверность 0.756				

Таблица 2.2.2 — Наивный байесовский классификатор

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.73	0.73	0.72	1192
1	0.71	0.75	0.72	1139
total	0.72	0.71	0.71	2331
Достоверность 0.730				

Таблица 2.2.3 — Случайный лес

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.71	0.74	0.72	1192
1	0.71	0.68	0.69	1139
total	0.71	0.71	0.71	2331
Достоверность 0.718				

Таблица 2.2.4 — Линейный метод опорных векторов

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.79	0.56	0.65	1192

1	0.64	0.85	0.73	1139
total	0.72	0.70	0.69	2331
Достоверность 0.689				

Метод Word2Vec (см. табл. 2.2.5, 2.2.6, 2.2.7 и 2.2.8).

Таблица 2.2.5 — Логистическая регрессия

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.77	0.77	0.77	1192
1	0.76	0.76	0.76	1139
total	0.86	0.86	0.86	2331
Достоверность 0.767				

Таблица 2.2.6 — Наивный байесовский классификатор

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.73	0.72	0.72	1192
1	0.71	0.72	0.71	1139
total	0.72	0.72	0.72	2331
Достоверность 0.728				

Таблица 2.2.7 — Случайный лес

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.73	0.74	0.75	1192
1	0.73	0.73	0.73	1139
total	0.73	0.73	0.74	2331
Достоверность 0.738				

Таблица 2.2.8 — Линейный метод опорных векторов

Класс	Мера точности	Мера полноты	Мера F1	Носитель меры
0	0.831	0.633	0.711	1092
1	0.691	0.864	0.771	1039

total	0.762	0.745	0.741	2131
Достоверность 0.743				

Свёрточная нейронная сеть

Достоверность 0.789.

Графики точности модели и функции потерь представлены на рис. 2.5 и 2.6 соответственно

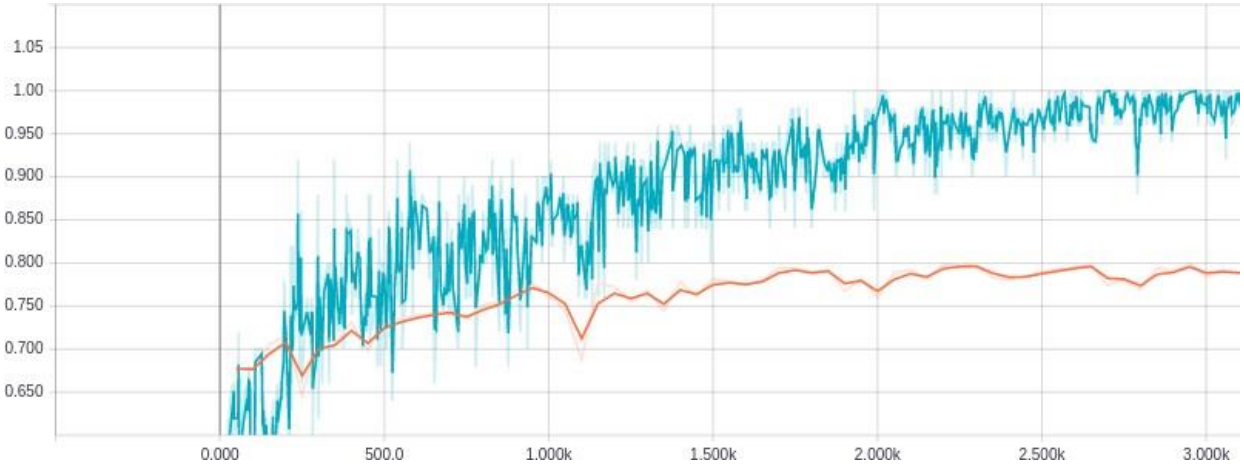

Рисунок 2.5 — Доля корректных прогнозов (accuracy): синий график – обучение, красный — проверка

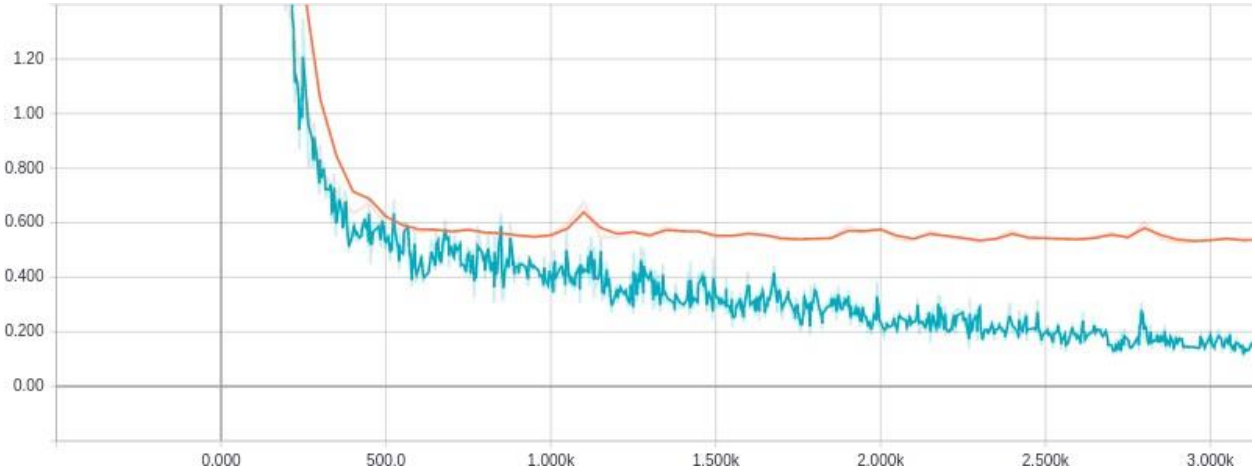

Рисунок 2.6 — Функция потерь: синий график – обучение, красный — проверка

Рекуррентная нейронная сеть с LSTM-блоками

Решающим в обучении модели оказался выбор функции минимизации градиентного спуска. Сначала был использован алгоритм RMSProp (root mean square propogation), идея которого заключается в масштабировании градиента.

Однако при прочих равных условиях (размере скрытого LSTM слоя и темпе обучения) применение алгоритма оптимизации Adam (который помимо идеи масштабирования градиента использует идею инерции) позволило достичь максимальной точности относительно уже реализованных методов в данной работе — 83.1%.

Графики точности модели и функции потерь представлены на рис. 2.7 и 2.8 соответственно.

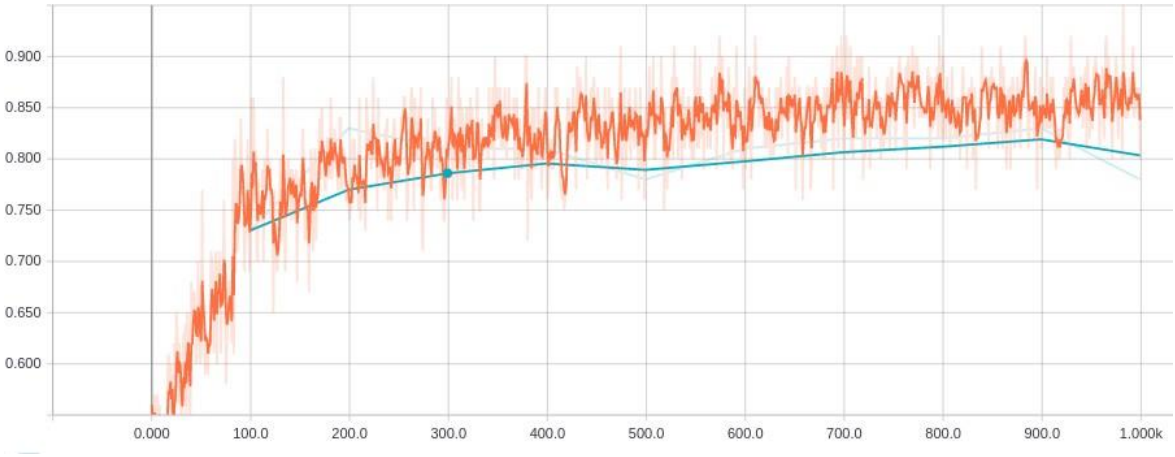

Рисунок 2.7 — Доля корректных прогнозов (accuracy): красный график–обучение, синий — проверка

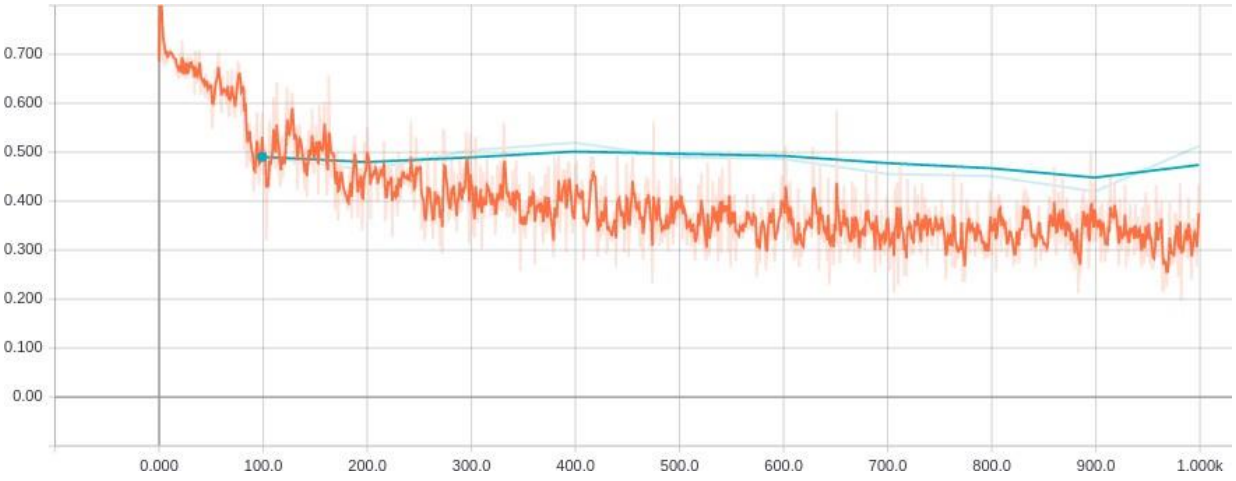

Рисунок 2.8 — Функция потерь: красный график – обучение, синий — проверка

2.4.Сравнение и оценка результатов

Для классических алгоритмов классификации применение мешка слов в свойстве модификации векторного представления слов достоверность предсказаний не превышала 74.4% (логистическая регрессия), минимальная достоверность составила 69.8% (линейный метод опорных векторов). Благодаря использованию модели Word2Vec удалось улучшить достоверность предсказаний почти для всех методов (кроме наивного байесовского классификатора): например, достоверность линейного метода опорных векторов стала 74.1%. Тем не менее самым эффективным бинарным классификатором оказалась логистическая регрессия: её достоверность с применением модели Word2Vec равна 75.6%.

При использовании свёрточных нейронных сетей с моделью Word2Vec удалось получить достоверность 79.8%. Самой эффективной архитектурой для анализа тональности текста оказалась рекуррентная нейронная сеть с LSTM-блоками. Её максимальная достоверность составила 83.1%. Полученные экспериментальные данные показывают более высокую эффективность работы глубоких нейронных сетей по сравнению с классическими алгоритмами для анализа тональности текста.

2.5.Морфологический анализ

Шаг 1. Выбор объекта.

Модуль контент-анализа на основе нейронных сетей

Шаг 2. Выбор основных характеристик объекта, которые выражаются отвлеченным понятием.

1. Фреймворк
2. Язык программирования

3. Среда разработки

Шаг 3. Указание всевозможных вариантов реализации характеристик, выбранных на шаге 2.

1. Фреймворк: Tensor Flow, Theano

2. Язык программирования: Python, C#

3. Среда разработки: PyCharm, Visual Studio

Критерии оценки:

1. Связанность компонентов

2. Сложность реализации

3. Временные затраты на разработку

Шаг 4. Рассмотрение различных полученных комбинаций.

Вариант 1:

Tensor Flow + Python + PyCharm

Вариант 1 отвечает выбранному нами критерию связанности компонентов. Среда разработки PyChram используется исключительно с языком программирования Python, а также фреймворк Tensor Flow реализовывается на языке программирования Python. Сложность реализации данного варианта сравнительно низка, так как язык программирования Python известен, среда разработки PyCharm также изучена и фреймворк Tensor Flow имеет прозрачную модульную архитектуру с множеством фронт-эндов за счёт чего достаточно прост в реализации. Таким образом временные затраты соответственно будут незначительными.

Вариант 2:

Tensor Flow + Python + Visual Studio

Вариант 2 отвечает выбранному нами критерию связанности компонентов. Среда разработки Visual Studio может быть использована с языком программирования Python, а также фреймворк Tensor Flow реализовывается на языке программирования Python. Сложность реализации

данного варианта сравнительно низка, так как язык программирования Python известен, среда разработки Visual Studio также изучена и фреймворк Tensor Flow имеет прозрачную модульную архитектуру с множеством фронт-эндов за счёт чего достаточно прост в реализации, однако, разработка в данной среде на выбранном языке программирования неудобна. Временные затраты будут средними.

Вариант 3:

Tensor Flow + C# + PyCharm

Вариант 3 не отвечает выбранному нами критерию связанности компонентов. Среда разработки PyChram используется исключительно с языком программирования Python и не может быть использована с языком программирования C#. Из этого следует, что данный вариант нам не подходит и дальнейший анализ данного варианта не имеет смысла.

Вариант 4:

Tensor Flow + C# + Visual Studio

Вариант 4 отвечает выбранному нами критерию связанности компонентов. Среда разработки Visual Studio используется с языком программирования C#, однако, фреймворк Tensor Flow не может быть реализован на данном языке программирования. Из этого следует, что данный вариант нам не подходит и дальнейший анализ данного варианта не имеет смысла.

Вариант 5:

Theano + Python + PyCharm

Вариант 5 отвечает выбранному нами критерию связанности компонентов. Среда разработки PyChram используется исключительно с языком программирования Python, а также фреймворк Theano реализовывается на языке программирования Python. Сложность реализации данного варианта сравнительно высока, так как язык программирования

Python известен, среда разработки PyCharm также изучена, но в коде фреймворка Theano сложно ориентироваться, его непросто отлаживать и проводить рефакторинг за счёт

чего реализация затруднена. Таким образом временные затраты будут умеренными.

Вариант 6:

Theano + Python + Visual Studio

Вариант 6 отвечает выбранному нами критерию связанности компонентов. Среда разработки Visual Studio может быть использована с языком программирования Python, а также фреймворк Theano реализовывается на языке программирования Python. Сложность реализации данного варианта сравнительно высока, так как язык программирования Python известен, среда разработки Visual Studuio также изучена, но в коде фреймворка Theano сложно ориентироваться, его непросто отлаживать и проводить рефакторинг за счёт чего реализация затруднена и разработка в данной среде на выбранном нами языке программирования неудобна. Таким образом временные затраты будут значительными.

Вариант 7:

Theano + C# + PyCharm

Вариант 7 не отвечает выбранному нами критерию связанности компонентов. Среда разработки PyChram используется исключительно с языком программирования Python и не может быть использована с языком программирования C#. Из этого следует, что данный вариант нам не подходит и дальнейший анализ данного варианта не имеет смысла.

Вариант 8:

Theano + C# + Visual Studio

Вариант 8 отвечает выбранному нами критерию связанности компонентов. Среда разработки Visual Studio используется с языком

программирования C#, однако, фреймворк Theano не может быть реализован на данном языке программирования. Из этого следует, что данный вариант нам не подходит и дальнейший анализ данного варианта не имеет смысла.

Шаг 5. Выбор оптимального варианта по обобщенным критериям

Оценки по частным критериям:

1. Отлично = 1,0
2. Очень хорошо = 0,75
3. Хорошо = 0,625
4. Удовлетворительно = 0,5
5. Посредственно = 0,25
6. Неудовлетворительно = 0

Критерий / №	Связанность компонентов	Сложность реализации	Временные затраты на разработку	**Итог**
Вариант 1 • Tensor Flow • Python • PyCharm	1,0	0,75	0,75	**2,5**
Вариант 2: • Tensor Flow • Python • Visual Studio	0,625	0,625	0,5	**1,75**
Вариант 5: • Theano • Python • PyCharm	1,0	0,5	0,5	**2,0**
Вариант 6:				

• Theano • Python • Visual Studio	0,625	0,25	0,25	**1,125**

Вывод.

После проведения морфологического анализа вариант 1 оказался оптимальным для использования в разработке модуля контент-анализа на основе нейронных сетей.

ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ

3.1 Средства реализации

Исходя из проведенного морфологического анализа был выбран язык программирования Python версии 3;

Благодаря проектированию системы и получению результатов анализа самой удачной стала рекуррентная нейронная сеть с LSTM-блоками дополнительными средствами реализации были выбраны библиотеки:

- Pandas для работы с данными;
- Scikit-Learn для токенизации, кросс-валидации и применения алгоритмов машинного обучения;
- PyMorphy2 для лемматизации русских слов.

3.2 Обучающая выборка

Создание обучающей выборки в задачах такого типа является сложным и долгим мероприятием. Необходимо большое количество времени и помощь большого количества людей (асессоров) для того, чтобы создать и разметить даже небольшую выборку. Поэтому была использованы готовая выборка, состоящей приблизительно из 225 тысяч размеченных твитов (положительная или отрицательная окраска).

Выборка содержит в себе не только тексты твитов и метки классов, но и большое количество дополнительной информации (даты публикаций, имена пользователей, количество ретвитов и т.д.). В контексте данной задачи эта информация не нужна, поэтому оставляем в датасете только тексты и метки.

Сохраняем обработанный датасет в новый файл *cleaned_data.csv*. в дальнейшем работая с ним.

3.3 Применение разработанного модуля

Анализ тональности текста, реализуемый данной системой, состоит из нескольких этапов. Сначала отрабатывает отдельный лингвистический модуль, автоматически производящий морфологический анализ текста, лемматизацию всей лексики и определяющий части речи каждого слова, его морфологические характеристики (падеж, лицо, число, активность-пассивность для глаголов), роль этого слова в предложении (для существительных: подлежащие, обстоятельство, дополнение; для глаголов: причастие, деепричастие, глагол; и др.), его тип (например, для существительных: физическое лицо, юридическое лицо, географическое название и др.). Затем все слова (существительные, глаголы, прилагательные и наречия) и некоторые словосочетания (коллокации) размечаются по заранее подготовленным словарным спискам тональной лексики. Каждому слову приписывается два атрибута, указывающие на тональность и/или силу тональности. Если слово не нашлось в списках тональной лексики, то оно считается нейтральным.

После этого запускается первичный синтаксический анализ: слова и словосочетания объединяются в тональные цепочки, в предложении выделяются субъект, предикат и объект, идентифицируются причастные и деепричастные обороты, подчинительные предложения, анафорические связи и пр. Естественно, не каждое предложение русского языка можно представить в виде триады субъект-предикат-объект. Учитываются также безличные, неопределенно-личные и обобщенно-личные предложения, предложения с нулевой формой глагола, сказуемые, выраженные неглагольной формой. На последнем этапе в предложении выделяется объект тональности и определяется его сентимент в зависимости от местоположения и роли этого

объекта в предложении. Результат работы програмы можно увидеть на рисунках 3.1 и 3.2, где зеленым цветом выделены позитивные слова, черным нейтральные и красным негативные.

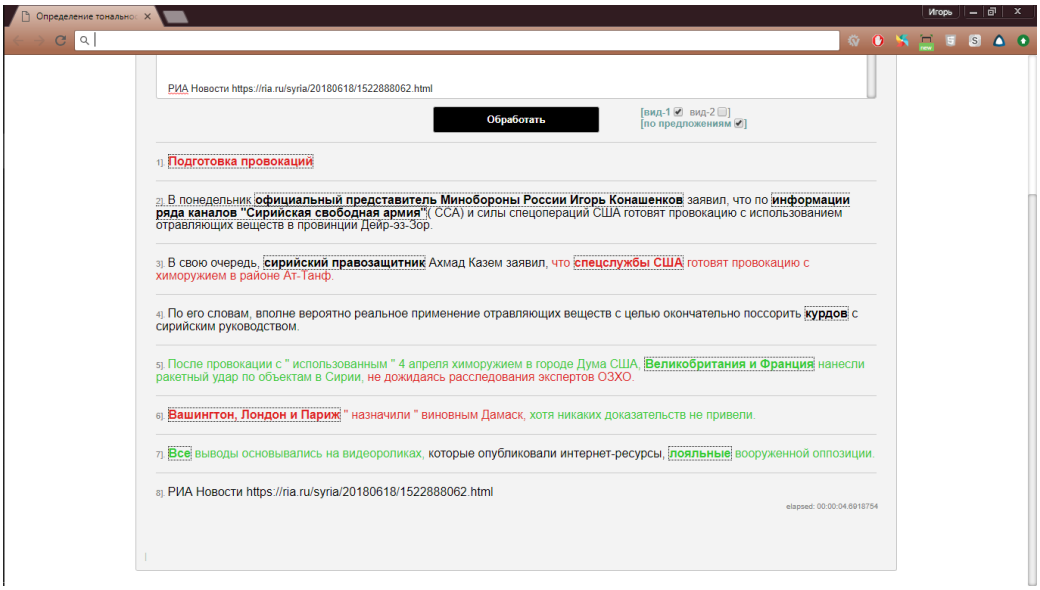

Рисунок 3.1 — Пример работы модуля

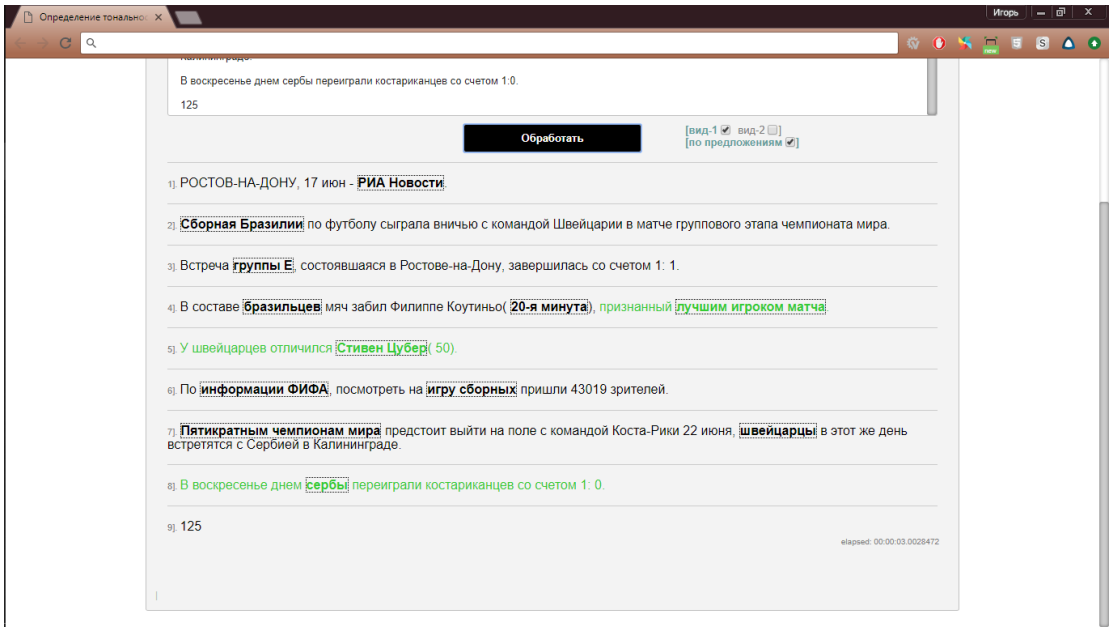

Рисунок 3.2 —Пример работы модуля

ЗАДАНИЕ ДЛЯ РАЗДЕЛА

«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К4Б	Чудину Игорю

Институт	ИК	Кафедра	ПИ
Уровень образования	Бакалавриат	**Направление/специальность**	09.03.04. Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, предоставленной в российских и иностранных научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах. Заработная плата руководителя по окладу - 16751,29 руб. Заработная плата бакалавра по окладу - 6976,22 руб.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценка потенциальных потребителей исследования, SWOT-анализ, анализ конкурентных решений
2. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоемкости и построение календарного графика, формирование бюджета

3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Оценка сравнительной эффективности исследования
Перечень графического материала *(с точным указанием обязательных чертежей)***:**	
1. Матрица SWOT *2. График Ганта*	

Дата выдачи задания для раздела по линейному графику		

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГТ	Петухов Олег Николаевич	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4Б	Чудин Игорь		

ГЛАВА 4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

Оценка коммерческого потенциала и перспективности проведения научных исследований

Потенциальные потребители результатов исследования

Данная работа нацелена на исследование систем анализа данных на базе Томского Политехнического Университета.

Предполагается использование данной системы для пользователей, связанных с рекламой, маркетингом, анализом и бизнесом с целью оценки позитивности содержания сообщений.

Анализ конкурентных технических решений

Детальный анализ конкурирующих разработок, существующих на рынке, необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам. Делая упор на слабые места конкурентов можно получить большое преимущество на рынке. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

С этой целью может быть использована вся имеющаяся информация о конкурентных разработках:

- технические характеристики разработки;
- конкурентоспособность разработки;
- уровень завершенности научного исследования (наличие макета, прототипа и т.п.);
- бюджет разработки;
- уровень проникновения на рынок;
- финансовое положение конкурентов, тенденции его изменения

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения. Далее будет представлена (таблица 2) оценочная

карта для сравнения конкурентных технических решений, $Бк_1$– «YandexDirect», $Бк_2$ – «EasySmartBiz».

Таблица 4.1 Оценочная карта

Критерии оценки	**Вес критерия**	**Баллы**			**Конкурентоспособность**		
		$Б_ф$	$Б_{к1}$	$Б_{к2}$	$К_ф$	$К_{к1}$	$К_{к2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Улучшение производительности труда заказчика	0.15	4	5	3	0,60	0,75	0,45
2. Функциональная мощность	0.1	5	5	4	0,50	0,50	0,40
3. Удобство в эксплуатации	0.15	3	4	3	0,45	0,60	0,45
4. Качество исполнения системы умного дома	0.1	4	3	2	0,40	0,30	0,20
5. Качество интеллектуального интерфейса	0.05	5	5	4	0,25	0,25	0,20
Экономические критерии оценки эффективности							
1.Конкурентоспособность разработки	0.1	4	5	4	0,40	0,50	0,40
2. Уровень востребованности среди потребителей	0.1	3	5	3	0,30	0,50	0,30
3. Цена	0.05	4	1	3	0,20	0,05	0,15
4. Финансирование научной разработки	0.1	3	3	3	0,30	0,30	0,30
5. Срок исполнения	0.1	4	5	3	0,40	0,50	0,30
Итого	**1**	**39**	**41**	**32**	**3,80**	**4,25**	**3,15**

Исходя из расчётов, сделанных выше, можно сделать вывод, что наша разработка имеет достаточно высокий уровень конкурентоспособности. Позиции конкурентов наиболее уязвимы в техническом развитии и ценовом диапазоне. Данные пункты определяют конкурентное преимущество нашей разработки.

SWOT-анализ

Для исследования внешней и внутренней среды проекта был проведен SWOT-анализ, который отражает сильные и слабые стороны разрабатываемого проекта.

Таблица 4.2 SWOT-анализ

	Сильные стороны научно-исследовательского проекта:	**Слабые сторон ы научно-исследовательского проекта:**
	С1. Удобство в эксплуатации (соответствует требования м потребителей). С2. Функциональная мощность (предоставляемые возможности). С3. Конкурентоспособно сть продукта. С4. Повышени е производительности труда. С5. Доступные технические средства разработки (.Net, ASP.Net MVC, Entity Framework, Microsoft SQL Server)	Сл1. Срок выхода на рынок. Сл2. Значительн ые временные и интеллектуальные затраты на реализацию. Сл3. Качеств о менеджмента.
Возможности: В1. Использован ие инновационной	Использование инновационной структуры ТПУ позволит повысить	Привлечение кадров из ТПУ увеличит штат сотрудников,

инфраструктуры ТПУ. В2. Публикации о проекте в тематических журналах. В3. Появление дополнительного спроса на новый продукт. В4. Повышение стоимости конкурентных разработок. В5. Привлечение специалистов из ТПУ для работы над проектом.	конкурентоспособность и ускорить выход на рынок. Возможно появление дополнительного спроса на новый продукт благодаря использованию доступных технических средств разработки.	работающих над проектом и позволит увеличить темпы работы над проектом. Публикация в журнале позволит познакомить целевую аудиторию с проектом.
Угрозы: У1. Отсутствие спроса на расширение разработки. У2. Отказ от технической поддержки проекта после внедрения. У3. Нехватка финансирования. У4. Развитая конкуренция разработчиков ИС.	Развитая конкуренция разработчиков может привести к снижению конкурентоспособности продукта. Отказ от технической поддержки может повлиять на мотивацию привлечения сотрудников в проект.	Отсутствие спроса на расширение разработки может замедлить срок выхода на рынок и понизить квалификацию научного труда. Нехватка финансирования также может затянуть срок выхода на рынок.

Таблица 4.3 Интерактивная матрица проекта

		Сильные стороны проекта					**Слабые стороны проекта**		
		С1	С2	С3	С4	С5	Сл1	Сл2	Сл3
Возможности проекта	В1	+	+	+	+	+	-	-	-
	В2	0	0	+	0	-	0	-	-
	В3	+	+	+	+	+	+	-	-
	В4	-	+	-	+	0	-	0	-
	В5	-	+	-	-	-	-	+	+
Угрозы проекта	У1	0	0	+	-	+	+	-	-
	У2	0	0	+	-	+	-	+	-
	У3	-	+	-	-	+	+	0	+
	У4	-	0	+	-	+	+	+	0

Определение возможных альтернатив проведения научных исследований

Морфологический подход основан на систематическом исследовании всех теоретически возможных вариантов, вытекающих из закономерностей строения (морфологии) объекта исследования. Синтез охватывает как известные, так и новые, необычные варианты, которые при простом переборе могли быть упущены. Путем комбинирования вариантов получают большое количество различных решений, ряд которых представляет практический интерес.

Морфологическая матрица приведена в таблице 5.

Таблица 4.4 Морфологическая матрица

	1	**2**
А. Среда разработки	Visual Studio	Eclipse
Б. База данных	Microsoft SQL Server Standard 2016	Oracle Database Standard Edition 2
В. Язык программирования	C#	Java
Г. Реализация	Web – приложение MVC5	Web – приложение

Для данной матрицы выберем три сочетания А1Б1В1Г1, А2Б2В2Г2.

Планирование научно-исследовательских работ

Структура работ в рамках научного исследования

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может варьироваться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей.

Перечень этапов и работ, распределение исполнителей по данным видам работ приведен в таблице 4.5.

Таблица 4.5 Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания.	Руководитель
Выбор направления исследований	2	Подбор материалов по теме	Бакалавр
	3	Изучение материалов по теме	Бакалавр
	4	Выбор направления	Руководитель, бакалавр
	5	Календарное планирование работ по теме	Бакалавр
Проектирование структуры и разработка ИС	6	Проектирование структуры ИС	Бакалавр
	7	Разработка ИС	Бакалавр
	8	Тестирование ИС	Бакалавр
Обобщение и оценка результатов	9	Оценка эффективности полученных результатов	Руководитель, бакалавр
Оформление отчета по НИР (комплекта документации по ОКР)	10	Составление пояснительной записки (эксплуатационно-технической документации)	Бакалавр

Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаях образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для выполнения

перечисленных в таблице 5 работ требуются специалисты:

бакалавр (Б);

научный руководитель (Р).

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях Тр, учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65 %.

Разработка графика проведения научного исследования

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Временные показатели проведения научного исследования представлены в таблице 4.6.

Таблица 4.6 Временные показатели проведения научного исследования

№ работ	Трудоёмкость работ						Исполнители		Длительность работ в рабочих днях T_{pi}		Длительность работ в календарных днях T_{ki}	
	tmin, чел-дни		tmax, чел-дни		$t_{ожi}$, чел-дни							
	Испол. 1	Испол. 2	Испол. 1	Испол. 2	Испол. 1	Испол. 2	Испол. 1	Испол. 2	Испол. 1	Испол. 2	Испол. 1	Испол. 2
1	4	4	6	6	4,8	4,8	Р	Р	4,8	4,8	6	6
2	8	8	12	12	9,6	9,6	Б	Б	9,6	9,6	12	12
3	14	14	20	20	16,4	16,4	Б	Б	8,2	12,3	10	10
4	7	7	12	12	9	9	Р, Б	Р, Б	3	3	4	4
5	4	4	10	10	6,4	6,4	Б	Б	2,1	2,1	3	3
6	23	22	25	26	23,8	23,6	Б	Б	11,9	13,5	14	14
7	18	20	20	23	18,8	21,2	Б	Б	9,4	16,2	11	13
8	1	1	2	2	1,4	1,4	Б	Б	0,7	0,7	1	1
9	3	3	5	5	3,8	3,8	Р, Б	Р, Б	1,3	1,3	2	2
10	17	17	23	23	19,4	19,4	Б	Б	9,7	9,7	12	12
Итого	Всего								60,7	73,2	74	75
	Руководитель								9,6	9,6	25	25
	Бакалавр								51,1	63,3	56	57

На основании таблицы 4.6 строится календарный план-график. График строится для максимального по длительности исполнения работ в рамках научно-исследовательского. План-график приведен в таблице 4.7.

Таблица 4.7 Календарный план-график

№ работ	Вид работ	Исполнители	T_{ki}, кал. дн.	Продолжительность выполнения работ					
				март		апрель		май	
				1	2	1	2	1	2
1	Составление и утверждение технического задания.	Руководитель	6						
2	Подбор материалов по теме	Руководитель	12						
3	Изучение материалов по теме	Бакалавр	10						
4	Выбор направления	Руководитель, бакалавр	4						
5	Календарное планирование работ по теме	Руководитель, бакалавр	3						
6	Проектирование структуры ИС	Бакалавр	16						
7	Разработка ИС	Бакалавр	14						
8	Тестирование ИС	Бакалавр	1						

9	Оценка эффективности и полученных результатов	Руководитель, бакалавр	2						
10	Составление пояснительной записки	Бакалавр	12						

Бюджет научно-технического исследования (НТИ)

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- материальные затраты НТИ;
- затраты на специальное оборудование для научных (экспериментальных) работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты научные и производственные командировки;
- контрагентные расходы;
- накладные расходы.

Расчет материальных затрат НТИ

Произведем расчет всех материалов, используемых при разработке проекта:

- приобретаемые со стороны сырье и материалы, необходимые для создания научно-технической продукции;
- покупные материалы, используемые в процессе создания научно-технической продукции для обеспечения нормального технологического процесса и для упаковки продукции или расходуемых на другие

производственные и хозяйственные нужды (проведение испытаний, контроль, содержание, ремонт и эксплуатация оборудования, зданий, сооружений, других основных средств и прочее), а также запасные части для ремонта оборудования, износа инструментов, приспособлений, инвентаря, приборов, лабораторного оборудования и других средств труда, не относимых к основным средствам, износ спецодежды и других малоценных и быстроизнашивающихся предметов;

- покупные комплектующие изделия и полуфабрикаты, подвергающиеся в дальнейшем монтажу или дополнительной обработке;

- сырье и материалы, покупные комплектующие изделия и полуфабрикаты, используемые в качестве объектов исследований (испытаний) и для эксплуатации, технического обслуживания и ремонта изделий – объектов испытаний (исследований);

В материальные затраты, помимо вышеуказанных, включаются дополнительно затраты на канцелярские принадлежности, диски, картриджи и т.п. Однако их учет ведется в данной статье только в том случае, если в научной организации их не включают в расходы на использование оборудования или накладные расходы. В первом случае на них определяются соответствующие нормы расхода от установленной базы. Во втором случае их величина учитывается как некая доля в коэффициенте накладных расходов.

Материальные затраты представлены в таблице 4.8.

Таблица 4.9 Материальные затраты

Наименование	Единица измерения	Количество		Цена за ед., руб.		Затраты на материалы, (**Зм**), руб.	
		Исп. 1	Исп. 2	Исп.1	Исп.2	Исп.1	Исп.2
Программное обеспечение							
Среда разработки	шт	1	1	29448	0	29448	0

База данных	шт	1	1	48255,38	39536,85	48255,38	39536,85
Офисные принадлежности							
Бумага для принтера А4	уп	1	1	150	150	150	150
Картридж для принтера	шт	1	1	500	500	500	500
Папка со скоросшивателем	шт	1	1	50	50	50	50
Итого				78403,38	40236,85	**78403,38**	**40236,85**

Расчет затрат на специальное оборудование для научных (экспериментальных) работ

Все расчеты по приобретению спецоборудования и оборудования, имеющегося в организации, но используемого для каждого исполнения конкретной темы, сводятся в таблице 10.

Таблица 4.10 Расчет бюджета затрат на приобретение спецоборудования для научных работ

№	Наименование оборудования	Кол-во единиц оборудования		Цена единицы оборудования, тыс. руб.		Общая стоимость оборудования, тыс. руб.	
		Испол. 1	Испол. 2	Испол. 1	Испол. 2	Испол. 1	Испол. 2
1.	Компьютер	1	1	17000	17000	19550	19550
2.	Монитор	1	1	7000	7000	8050	8050
3.	Принтер	1	1	5000	5000	5750	5750
Итого						**33350**	**33350**

Основная заработная плата исполнителей темы

Рассчитаем основную заработную плату работников, непосредственно занятых выполнением НТИ, (включая премии, доплаты) и дополнительную заработную плату:

Таблица 4.11 Баланс рабочего времени

Показатели рабочего времени	**Руководитель**	**Студент**
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	107	107
Потери рабочего времени - отпуск - невыходы по болезни	24	24
Действительный годовой фонд рабочего времени	234	234

Расчёт основной заработной платы приведён в табл. 4.12.

Таблица 4.12 Расчёт основной заработной платы

Исп.	Исполнители	Разряд	Зтс, руб.	*k*р	Зм, руб	Здн, руб.	Тр, раб. дн.	Зосн, руб.
Исп. 1	Руководитель	Ст. преп.	16751,29	1,3	31576,18	1511,34	9,6	14508,86
	Бакалавр	1	6976,22	1,3	13150,17	629,41	51,1	32162,85
	Итого							**46671,71**
Исп. 2	Руководитель	Ст. преп.	16751,29	1,3	31576,18	1511,34	9,6	14508,86
	Бакалавр	1	6976,22	1,3	13150,17	629,41	63,3	39841,65
	Итого							**54350,51**

Дополнительная заработная плата исполнителей темы

Таблица 4.13 Расчёт дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.		$k_{доп}$	Дополнительная заработная плата, руб.	
	Испол.1	Испол.2		Испол.1	Испол.2
Руководитель	14508,86	14508,86	0,12	1741,06	1741,06
Бакалавр	32162,85	39841,65		3859,54	4781
Итого				**5600,60**	**6522,06**

Отчисления во внебюджетные фонды (страховые отчисления

Таблица 4.14 Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.		Полная заработная плата, руб.	
	Испол. 1	Испол.2	Испол.1	Испол.2
Руководитель	14508,86	14508,86	16249,92	16249,92
Бакалавр	32162,85	39841,65	36022,39	44622,65
Коэффициент отчислений во внебюджетные фонды	0,271			
Итого				
Исполнение 1	**14165,79**			
Исполнение 2	**16496,46**			

Расчет затрат на научные и производственные командировки

На данном этапе в научных и производственных командировках нет необходимости.

Контрагентные расходы

На данном этапе невозможно оценить влияние контрагентных расходов на проект.

4.3.4.8 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д.

Исполнение 1 = (78403,38+33350+46671,71+5600,60+14165,79) * 0,16 = 178191,48 * 0,16 = 28510,64

Исполнение 2 = (40236,85+33350+54350,51+6522,06+16496,46) * 0,16 = 150955,88 * 0,16 = 24152,94

4.3.4.9 Формирование бюджета затрат научно-исследовательского проекта

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 4.15.

Таблица 4.15. Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.		Примечание
	Испол.1	Испол.2	
1. Материальные затраты НТИ	78403,38	40236,85	Пункт 3.4.1
2. Затраты на специальное оборудование для научных (экспериментальных) работ	33350	33350	Пункт 3.4.2
3. Затраты по основной заработной плате исполнителей темы	46671,71	54350,51	Пункт 3.4.3
4. Затраты по дополнительной заработной плате исполнителей темы	5600,60	6522,06	Пункт 3.4.4
5. Отчисления во внебюджетные фонды	14165,79	16496,46	Пункт 3.4.5
6. Затраты на научные и производственные	0	0	Пункт 3.4.6

командировки			
7. Контрагентские расходы	0	0	Пункт 3.4.7
8. Накладные расходы	28510,64	24152,94	16 % от суммы ст. 1-7
9. Бюджет затрат НТИ	206702,12	175108,82	Сумма ст. 1- 8

Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования (см. табл. 15). Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносится финансовые значения по всем вариантам исполнения. Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разах (значение меньше единицы, но больше нуля).

Таблица 4.16 Сравнительная оценка характеристик вариантов исполнения проекта

Критерии / Объект исследования	Весовой коэффициент параметра	Испол. 1	Испол. 2
1. Улучшение производительности труда заказчика	0,25	5	4
2. Функциональная мощность	0,20	4	3
3. Удобство в эксплуатации	0,25	5	4
4. Потребность в ресурсах памяти	0,15	5	3
5. Надежность	0,15	4	4
ИТОГО	1		

*I р-исп1= 5*0,25+4*0,20+5*0,25+5*0,15+4*0,15 = 4,65;*

*I р-исп2= 4*0,25+3*0,20+4*0,25+3*0,15+4*0,15 = 3,*

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта. Сравнительная эффективность разработки представлена в таблице 17.

Таблица 4.17 Сравнительная эффективность разработки

№ п/п	Показатели	Испол. 1	Испол.2
1	Интегральный финансовый показатель исполнения	1	0,86
2	Интегральный показатель ресурсоэффективности исполнения	4,65	3,65
3	Интегральный показатель эффективности вариантов исполнения	4,65	4,24
4	Сравнительная эффективность вариантов исполнения	1,09	0,91

Общий вывод по разделу:

В результате работы по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» выявили и сравнили два варианта исполнения научно-исследовательской работы. Бюджет затрат первого варианта исполнения равен 206702,12 рублей, второго – 175108,82. Произвели сравнительную оценку эффективности разработки и исходя из полученных результатов можно сделать вывод, что наиболее эффективным вариантом решения поставленной в бакалаврской работе технической задачи с позиции финансовой и ресурсной эффективности является 1 вариант исполнения – использование среды разработки Visual Studio.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К4Б	Чудину Игорю

Школа	ИШИТР	Отделение	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04. Программная инженерия

Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Разработка модуля для информационной системы на рабочем месте, при помощи персонального компьютера, монитора, клавиатуры и компьютерной мыши.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Профессиональная социальная ответственность. 1.1. Анализ вредных факторов проектируемой производственной среды. 1.2. Анализ опасных факторов проектируемой произведённой среды.	Анализ вредных факторов: - Параметры микроклимата по СанПиН 2.2.2/ 2.4.1340-03 - Освещенность рабочего места по СанПиН 2.2.1/2.1.1.1278-03 - Уровень шума по СанПиН 2.2.4.3359-16 - Умственное перенапряжение по ТОИ Р-45-084-01 Анализ опасных факторов: - Опасность поражения электрическим током по ГОСТ Р 50571. 17-2000 - Короткое замыкание - Статическое электричество

2. Экологическая безопасность:	Анализ негативного воздействия на окружающую природную среду: утилизация компьютеров и другой оргтехники. В том числе мусорные отходы (бумага). Люминесцентные лампы.
3. Безопасность в чрезвычайных ситуациях:	Вероятно-возможные ЧС: - пожар Мероприятия по предотвращению ЧС.
4. Правовые и организационные вопросы обеспечения безопасности:	Организация рабочего места согласно ГОСТ 12.2.032-78 Анализ конструкции рабочей мебели для работ сидя, согласно ГОСТ 12.2.061-81 «Трудовой кодекс РФ» от 30.12.2001 N 197-ФЗ

Дата выдачи задания для раздела по линейному графику	01.03.2018

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева Ирина Ивановна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4Б	Чудин Игорь		

ГЛАВА 5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Выпускная квалификационная работа по проектированию и реализации информационной системы анализа данных выполнялась на кафедре Программной Инженерии в одном из кабинетов Кибернетического центра Томского Политехнического Университета. Проектируемое рабочее место представляет собой офисное помещение, в котором будет работать инженер-программист.

В данной работе освещен комплекс мер организационного, правового, технического и режимного характера, которые минимизируют негативные последствия разработки информационной системы, а также рассматриваются вопросы техники безопасности, охраны окружающей среды и пожарной профилактики, даются рекомендации по созданию оптимальных условий труда.

Специфика и режим работы разработчика характеризуются значительным умственным напряжением, сильной нагрузкой на зрительный аппарат, неподвижностью и напряженностью в шейно-грудном и поясничном отделах позвоночника, что приводит к появлению усталости изменению функционального состояния центральной нервной системы, появлению болей в запястьях, локтевых суставах, кистях, пальцах рук и спине. При длительной работе за экраном монитора появляются болезненные ощущения в глазах и головная боль.

Разработка информационной системы никаким образом не оказывает отрицательного воздействия на общество и окружающую среду, но в процессе работы специалиста с информационной системой возможно образование твердых отходов, таких как бумага, батарейки, лампочки, использованные картриджи, отходы от продуктов питания и личной гигиены, отходы от канцелярских принадлежностей и т.д.

5.1 Техногенная безопасность.

По природе возникновения вредные и опасные производственные факторы делятся на 4 группы:

- физические;
- химические;
- психофизиологические;
- биологические.

В нашем случае биологические и химические факторы существенного влияния на состояние здоровья исполнителей не оказывают, то подробнее рассмотрим лишь физические и психофизиологический факторы.

Единственным фактором, относящимся к физически опасным, является опасность поражения электрическим током. В качестве же вредных производственных факторов, которые имеют место при работе с компьютером, были выделены следующие позиции:

К вредным производственным факторам, при работе с компьютером следует отнести:

1) повышенный уровень электромагнитных излучений, основными источниками которых является электроннолучевая трубка монитора компьютера; [5]

2) отклонение показателей микроклимата [3]

3) повышенный уровень шума, источниками которого являются вентиляторы внутри системного блока и блока питания компьютера, накопители на жестких и магнитных дисках, светильники люминесцентных ламп и др. [5]

4) недостаточная освещённость рабочей зоны [6]

5.1.1 Уровень электромагнитных излучений

Как любые электрические приборы, видеотерминалы (ВДТ) и системные блоки производят электромагнитное излучение, воздействие этого изучение на организм человека напрямую зависит от напряжённостей электрического, магнитного поля, от потока энергии, частоты колебаний, а также от размера облучаемого тела.

При воздействии электромагнитных полей низкой напряжённости нарушения, возникающие в организме человека, носят обратимый характер.

Однако если напряжённость магнитных полей выше предельно допустимого уровня, то страдают нервная и сердечно-сосудистая системы, органы пищеварения, а также ухудшаются некоторые биологические показатели крови.

Большая часть электромагнитных излучений происходит не от экрана монитора, а от видеокабеля и системного блока. В портативных компьютерах практически всё электромагнитное излучение идет от системного блока, располагающегося под клавиатурой. Современные машины выпускаются

заводом-изготовителем со специальной металлической защитой внутри системного блока для уменьшения фона электромагнитного излучения.

Согласно [5] на расстоянии 50см вокруг ВДТ напряженность электромагнитного поля по электрической составляющей должна быть не более:

25 В/м, если частота находится в диапазоне 5 Гц ÷ 2 кГц

2,5 В/м, если частота находится в диапазоне 2 кГц ÷ 400кГц

Плотность магнитного потока не должна превышать:

250 нТл, если частота находится в диапазоне 5 Гц ÷ 2 кГц

25 нТл, если частота находится в диапазоне 2 кГц ÷ 400кГц

Возможные способы защиты от ЭМП:

Основной подход – увеличить расстояние от источника, экран видеомонитора не должен находится ближе 50 см от пользователя;

Применение приэкранного фильтра, специального экрана, а также других средств индивидуальной защиты, которые прошли испытание в аккредитованных лабораториях и которые имеют соответствующий гигиенический сертификат.

5.1.2 Показателей микроклимата

Проанализируем микроклимат на рабочем месте. Микроклимат производственных помещений характеризуется следующими параметрами: температурой, относительной влажностью, скоростью движения воздуха. Все эти параметры влияют на организм человека как сами по себе, так и в комплексе. Они во многом определяют самочувствие. Оптимальные значения характеристик микроклимата установлены в соответствии с [3] и отображены в таблице 5.1.

По степени физической тяжести работа инженера-программиста относится к лёгкой физической работе категории I а, с энергозатратами организма до 120 Дж/с, т.к. работа проводилась сидя, не требуя систематического физического напряжения.

Таблица 5.1 Оптимальные значения характеристик микроклимата

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, ◦С	Температура поверхностей, ◦С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	1а(до 139)	22-24	21-25	60-40	0,1
Теплый	1а(до 139)	23-25	21-26	60-40	0,1

Допустимые величины показателей микроклимата устанавливаются в случаях, когда по технологическим требованиям, техническим и экономически обоснованным причинам не могут быть обеспечены оптимальные величины.

Таблица 5.2 Допустимые значения микроклимата рабочего стола.

Период года	Категория работ	Температура воздуха, ◦С		Температура поверхностей, ◦С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
		Ниже опт	Выше опт			Ниже опт	Выше опт
Холодный	1а(до 139)	20-21,9	24,2-25	19-26	15-75	0,1	
Теплый	1а(до 139)	21-22,9	25,1-28	20,29	15-75	0,1	0,2

Параметры микроклимата помещения, регулирующиеся системой центрального отопления, а также приточно-вытяжной вентиляцией, имеют следующие значения:

влажность 40%,

скорость движения воздуха 0,1 м/с,

температура летом 20-25°С, зимой 15-18°С,

Что соответствует требованиям [3].

Если говорить о мероприятиях по оздоровлению воздушной среды, то в производственном помещении к ним относится правильная организация вентиляции и кондиционирования воздуха, а также отопление помещений.

Вентиляция должна осуществляться как естественным, так и механическим путём. В рабочем помещении необходима подача следующего объёма наружного воздуха: при объёме помещения до 20м3 на человека – не менее 30м3 в час на человека; при объёме помещения более 40м3 на человека

и отсутствии выделения вредных веществ допускается естественная вентиляция.

В аудитории принудительная вентиляция отсутствует. Но имеется естественная, т.е. воздух поступает и удаляется через окна, двери, щели.

Весомый недостаток естественной вентиляции в том, что воздух поступает в помещение без очистки и нагревания. Естественная вентиляция допускается в том случае, если на одного работающего приходится не менее 40м3 всего объема воздуха в помещении. Объём воздуха на одного человека в аудиториях КЦ — 28,88м3, следовательно, необходимо наличие принудительной вентиляции.

В зимнее время в помещении должна быть система отопления. Она обеспечивает достаточное, постоянное и равномерное нагревание воздуха. В помещениях с повышенными требованиями к чистоте воздуха должно использоваться водяное отопление. В аудиториях используется водяное отопление со встроенными нагревательными элементами и стояками.

5.1.3 Освещённость рабочей зоны

Недостаточная освещенность пагубно влияет на зрительный аппарат, то

есть снижает зрительную работоспособность, также освещенность рабочей зоны влияет на психику человека, эмоциональное состояние, может вызывать усталость центральной нервной системы, которая возникает в результате приложения дополнительных усилий для опознания четких или сомнительных сигналов.

Для оптимизации условий труда большую роль играет освещение рабочих мест [6]. Организация освещённости рабочих мест должно выполнить два требования: обеспечить различаемость рассматриваемых предметов и уменьшить напряжение и утомляемость органов зрения. Производственное освещение должно быть устойчивым и равномерным, иметь правильное направление, исключать слепящее действие и образование резких теней.

Основным качественным показателям световой среды является коэффициент пульсации освещенности (Кп). Для рабочих мест с ПЭВМ этот показатель не должен превышать 5%. Оптимальная яркость экрана дисплея составляет 75–100 кд/м2. При такой яркости экрана, а также яркости поверхности стола в пределах от 100 до 150 кд/м2 обеспечивается работоспособность зрительного аппарата на уровне 80–90 % и сохраняется постоянный размер зрачка на допустимом уровне 3–4 мм. Местное освещение не должно создавать блики на поверхности экрана и не должно увеличивать освещенность экрана

ПЭВМ более, чем 300 лк. Следует ограничивать прямую и отраженную блёскость от любых источников освещения.

В лаборатории, где проводится ВКР, используется смешанное освещение, т.е. сочетание естественного и искусственного освещения. Естественным освещением является освещение через окна.

Искусственное освещение используется при недостаточном естественном освещении. В данном помещении используется общее искусственное освещение. Помещение, где проводится ВКР, освещается 3 светильниками, в каждом из которых установлено 4 люминесцентных лампы типа ЛБ-40. Светильники расположены равномерно по всей площади потолка в ряд, создавая при этом равномерное освещение рабочих мест. Световой поток каждой из ламп в помещении свидетельствует о соблюдении норм освещенности.

Следует ограничивать отраженную блёскость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м2 и яркость потолка, при применении системы отраженного освещения, не должна превышать 200кд/м2.

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛБ. Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении ПЭВМ.

Для освещения помещений с ПЭВМ следует применять светильники серии ЛПО36 с зеркализованными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Применение светильников без рассеивателей и экранирующих решеток не допускается. Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м2, защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

В помещении три оконных проема. КЕО при совмещенном освещении и боковом естественном освещении для данного типа помещений составляет 0,7.

Уровень искусственного освещения должен быть не менее 300 лк.[6]

Таблица 5.3 Параметры систем естественного и искусственного освещения на рабочих местах

Наименование рабочего места	Тип светильника и источника света	Коэффициент Естественной освещенности, КЕО,%		Освещенность при совмещенной системе, лк	
		Фактически	Норм.значение	Фактически	Норм.значение
Помещение для работы с ПЭВМ	ОДР ЛБ-40	---	0,7	1021 лк	300÷500 лк

5.1.4 Уровень шума на рабочем месте

Одним из важнейших параметров, которые наносят большой ущерб здоровью и резко снижают производительность труда, является шум. Шум может создаваться чем угодно, будь это работающее оборудование, установки кондиционирования воздуха, преобразователи напряжения, работающие осветительные приборы дневного света, или шум, проникающий извне.

В ходе исследований установлено, что шум и вибрация оказывают пагубное воздействие на организм человека. Действие шума различно: он затрудняет разборчивость речи, снижает работоспособность, повышает утомляемость, вызывает изменения в органах слуха человека. Шум воздействует на весь организм человека, а не только на органы слуха. Отмечается ослабление внимания, ухудшение памяти, снижение реакции, увеличение числа ошибок при работе.

Производственные помещения, в которых для работы используются ПЭВМ, не должны находиться по соседству с помещениями, в которых уровень шума и вибрации превышают нормируемые значения.

Допустимый уровень звукового давления, звука и эквивалентные уровни звука на рабочих местах должны отвечать требованиям СанПиН 2.2.4.3359-16 [4].

При выполнении основной работы на ПЭВМ уровень шума на рабочем месте не должен превышать 50 дБА.

5.1.5 Умственное перенапряжение

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности.

Виды трудовой деятельности делятся на 3 группы: группа А-работа по считыванию информации с экрана ВДТ с предварительным запросом, группа Б-работа по вводу информации, группа В-творческая работа в режиме диалога с ПЭВМ.

Для видов деятельности устанавливается 3 категории тяжести и напряженности работы с ПЭВМ, которые определяются: для группы А-по суммарному числу считываемых знаков за рабочую смену, но не более 60 000 знаков за смену; для группы Б-по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40 000 знаков за смену; для группы В-по суммарному времени непосредственной работы с ПЭВМ за рабочую смену, но не более 6 ч. за смену.

В зависимости от категории трудовой деятельности и уровня нагрузки за рабочую смену при работе с ПЭВМ устанавливается суммарное время регламентированных перерывов.

Категория работ по тяжести и напряженности по ТОИ Р 45-084-01 представлена в таблице 5.

Таблица 5.4. Категория работ по тяжести и напряженности по ТОИ Р 45-084-01

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин	
	Группа А, кол-во знаков	Группа Б, кол-во знаков	Группа В, ч	При 8-ми часовой смене	При 12-ти часовой смене
III	До 60 000	До 40 000	До 6	90	140

При 8-ми часовой работе на ПЭВМ регламентированные перерывы следует устанавливать через 1,5-2 часа от начала сеанса и через 1,5-2 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час обучения.

При 12-часовом сеансе регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-ми часовом сеансе, а в течение последних 4-х часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

5.2 Безопасность в чрезвычайных ситуациях.

5.2.1 Пожарная безопасность помещения

Согласно ГОСТ Р 50571.17-2000 [8], в зависимости от характеристики используемых в производстве веществ и их количества, по пожарной и взрывной опасности помещения подразделяются на категории А, Б, В, Г, Д.

Наличие в аудитории 204-КЦ деревянных изделий (столы, шкафы), электропроводов напряжением 220В, а также применение электронагревательных приборов с открытыми нагревательными элементами– паяльниками дает право отнести помещение по степени пожаро- и взрыво-

безопасности к категории В.

Необходимо предусмотреть ряд профилактических мероприятий технического, эксплуатационного, организационного плана.

В качестве возможных причин пожара можно указать следующие:

1)короткие замыкания;

2)опасная перегрузка сетей, которая ведет за собой сильный нагрев токоведущих частей и загорание изоляции;

3)нередко пожары происходят при пуске оборудования после ремонта.

Для предупреждения пожаров от коротких замыканий и перегрузок необходимы правильный выбор, монтаж и соблюдение установленного режима эксплуатации электрических сетей, дисплеев и других электрических средств

автоматизации.

Следовательно, необходимо предусмотреть ряд профилактических мероприятий технического, эксплуатационного, организационного плана.

5.2.2 Возможные причины загорания

Причиной возгорания может быть:

1)неисправность токоведущих частей установок;

2)работа с открытой электроаппаратурой;

3)короткие замыкания в блоке питания или высоковольтном блоке дисплейной развертки;

4)несоблюдение правил пожарной безопасности;

5)наличие горючих компонентов: документы, двери, столы, изоляция кабелей и т.п.

5.2.3 Мероприятия по устранению и предупреждению пожаров

Для предупреждения возникновения пожара необходимо соблюдать следующие правила пожарной безопасности:

1)исключение образования горючей среды (герметизация оборудования, контроль воздушной среды, рабочая и аварийная вентиляция);

2)применение при строительстве и отделке зданий несгораемых или трудно сгораемых материалов.

Необходимо в аудитории проводить следующие пожарно-профилактические мероприятия:

1)организационные мероприятия, касающиеся технического процесса с учетом пожарной безопасности объекта;

2)эксплуатационные мероприятия, рассматривающие эксплуатацию имеющегося оборудования;

3)технические и конструктивные, связанные с правильным размещением и монтажом электрооборудования и отопительных приборов.

Организационные мероприятия:

1)противопожарный инструктаж обслуживающего персонала;

2)обучение персонала правилам техники безопасности;

3)издание инструкций, плакатов, планов эвакуации.

Эксплуатационные мероприятия:

1)соблюдение эксплуатационных норм оборудования;

2)обеспечение свободного подхода к оборудованию;

3)содержание в исправности изоляции токоведущих проводников.

Технические мероприятия:

1) Соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения. В аудитории 204-КЦ имеется углекислотный огнетушитель типа ОУ–2, установлен рубильник, обесточивающий всю аудиторию, на двери аудитории приведен план эвакуации в случае пожара, и на досягаемом расстоянии находится пожарный щит (2 этаж КЦ). Если возгорание произошло в электроустановке, для его устранения должны использоваться углекислотные огнетушители типа ОУ–2.

2) Профилактический осмотр, ремонт и испытание оборудования. Кроме устранения самого очага пожара, нужно своевременно организовать эвакуацию людей.

5.2.4 Электробезопасность

В этом разделе нас интересует статическое электричество, которое возникает в результате процессов перераспределения электронов и ионов, когда происходит соприкосновение двух поверхностей неоднородных жидких, либо твердых веществ, на которых образуется двойной электрический слой.

Разделении поверхностей означает разделение зарядов этого слоя, а значит между разделенными поверхностями возникает разность потенциалов и образуется электрическое поле.

В помещении статическое электричество часто возникает при прикосновении человека к элементам ЭВМ. Разряды не представляют опасность для пользователей, но они могут привести проблемам с ЭВМ.

Чтобы снизить величины возникающих зарядов статического электричества покрытие полов в помещении выполняется из однослойного линолеума.

При работе с электроприборами крайне важно соблюдать технику безопасности.

Под техникой безопасности подразумевается система организационных мероприятий и технических средств, которые направлены на предотвращения воздействия на пользователя вредных и опасных производственных факторов.

Электрические установки представляют серьезную потенциальную опасность для пользователя, это еще усугубляется тем фактом, что органы чувств человека не могут обнаружить наличие электрического напряжения на расстоянии.

Опасность поражения человека электрическим током напрямую зависит от условий в помещении. Риск поражения возрастает при следующих

условиях: повышенная влажность (относительная влажность воздуха превышает 75%), высокая температура (более 35°С), наличие токопроводящей пыли, токопроводящих полов, а также возможности одновременного соприкосновения к металлическим элементам, имеющим соединение с землей, и металлическим корпусом электрооборудования. Следовательно, работа может проводиться исключительно в помещениях без повышенной опасности, при этом существует опасность электропоражения:

1) при прикосновении к токоведущим частям, например, во время ремонта ПЭВМ;

2) при прикосновении к нетоковедущим частям, которые оказались под напряжением (при нарушении изоляции токоведущих частей ПЭВМ);

3) при соприкосновении с полом, стенами, оказавшимися под напряжением;

4) имеется опасность короткого замыкания в высоковольтных блоках: блоке питания и блоке дисплейной развёртки.

Аудитории КЦ, в которых проводились работы, по опасности электропоражения не относятся к помещениям повышенной опасности.

В лабораториях используются приборы, потребляющие напряжение 220В переменного тока с частотой 50Гц. Это напряжение опасно для жизни, поэтому обязательны следующие меры предосторожности:

1)перед началом работы необходимо убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;

2)при обнаружении неисправности оборудования и приборов, необходимо не делая никаких самостоятельных исправлений сообщить ответственному за оборудование;

3)запрещается загромождать рабочее место лишними предметами. При возникновении несчастного случая следует немедленно освободить пострадавшего от действия электрического тока и, вызвав врача, оказать ему необходимую помощь.

5.3 Экологическая безопасность

Научно-технический прогресс, увеличивает возможности человека воздействовать на окружающую среду, это создает условия для возникновения

экологического кризиса. При этом развитие технологий открывает и новые пути поддержания природной среды и предлагает новые варианты преодоления уже существующих проблем.

Под окружающей средой будем понимать совокупность природы и среды созданной человеком.

Защита окружающей среды - это комплексная проблема, требующая усилий всего человечества. Наиболее активной формой защиты окружающей среды от вредного воздействия выбросов промышленных предприятий является полный переход к безотходным и малоотходным технологиям и производствам.

Это потребует решения целого комплекса сложных технологических, конструкторских и организационных задач, основанных на использовании новейших научно-технических достижений.

5.3.1 Отходы

Основные виды загрязнения литосферы – твердые бытовые и промышленные отходы.

При рассмотрении влияния ПЭВМ на атмосферу, гидросферу и литосферу выявлены особо вредные выбросы согласно ГОСТ Р 51768-2001 «Ресурсосбережение. Обращение с отходами». В случае выхода из строя компьютера, они списываются и отправляются на специальный склад, который при необходимости принимает меры по утилизации списанной техники и комплектующих.

Люминесцентные лампы в случае нарушения целостности корпуса отслуживших свой срок изделий выделяются пары ртути. Лампы по окончании этого срока положено сдавать на специальные предприятия, где они подлежат дальнейшей утилизации, суть которой состоит в сборе и нейтрализации веществ, содержащих ртуть.

Защита почвенного покрова и недр от твердых отходов реализуется за счет сбора, сортирования и утилизации отходов и их организованного захоронения.

5.4 Организационные мероприятия обеспечения безопасности

При организации рабочего места необходимо учитывать требования безопасности, промышленной санитарии, эргономики, технической эстетики.

Невыполнение этих требований может привести к получению работником производственной травмы или развитию у него профессионального заболевания. Согласно требований [9,10,11] при организации работы на ПЭВМ должны выполняться следующие условия:

- персональный компьютер(ПК), и соответственно рабочее место должно располагаться так, чтобы свет падал сбоку, лучше слева;

- расстояние от ПК до стен должно быть не менее 1 м, поэтому по возможности следует избежать расположение рабочего места в углах помещения либо лицом к стене;

- ПК лучше установить так, чтобы, подняв глаза от экрана, можно было увидеть какой-нибудь удаленный предмет в помещении или на улице. Перевод взгляда на дальнее расстояние является одним из наиболее эффективных способов разгрузки зрительного аппарата при работе на ПК;

- при наличии нескольких компьютеров расстояние между экраном одного монитора и задней стенкой другого должно быть не менее 2 м, а расстояние между боковыми стенками соседних мониторов – не менее 1,2 м;

- окна в помещениях с ПЭВМ должны быть оборудованы регулируемыми устройствами (жалюзи, занавески, внешние козырьки и т.д.);

- монитор, клавиатура и корпус компьютера должны находиться прямо перед оператором; высота рабочего стола с клавиатурой должна составлять 680 – 800 мм над уровнем стола; а высота экрана (над полом) –900–1280см;

- монитор должен находиться от оператора на расстоянии 60 – 70 см на 20 градусов ниже уровня глаз;

- пространство для ног должно быть: высотой не менее 600 мм, шириной не менее 500 мм, глубиной не менее 450 мм. Должна быть предусмотрена подставка для ног работающего шириной не менее 300 мм с регулировкой угла наклона 0-20 градусов;

- рабочее кресло должно иметь мягкое сиденье и спинку, с регулировкой сиденья по высоте, с удобной опорой для поясницы;

- Положение тела пользователя относительно монитора должно соответствовать направлению просмотра под прямым углом или под углом 75 градусов. Правильная поза и положение рук оператора являются весьма важными для исключения нарушений в опорно-двигательном аппарате и возникновения синдрома постоянных нагрузок.

Согласно СанПиНу 2.2.2.542-96 при 8-ми часовой рабочей смене на ВДТ и ПЭВМ перерывы в работе должны составлять от 10 до 20 минут каждые два часа работы.

5.5 Особенности законодательного регулирования проектных решений.

5.5.1 Специальные правовые нормы трудового законодательства

Условия труда, созданные при выполнении ВКР, не являются опасными или вредными для здоровья и не несут угрозу экологической безопасности. График работы не нарушался. В лаборатории регулярно проводились организационно-технические мероприятия, например, первичный инструктаж по технике безопасности и целевой инструктаж по проведению работ.

5.5.2 Организационные мероприятия при компоновке рабочей зоны

При организации рабочего места необходимо учитывать требования безопасности , эргономики, промышленной санитарии, технической эстетики. Невыполнение этих требований может привести к получению работником производственной травмы или развитию у него профессионального заболевания.

Организация работы на ПЭВМ показана на рисунке 1.

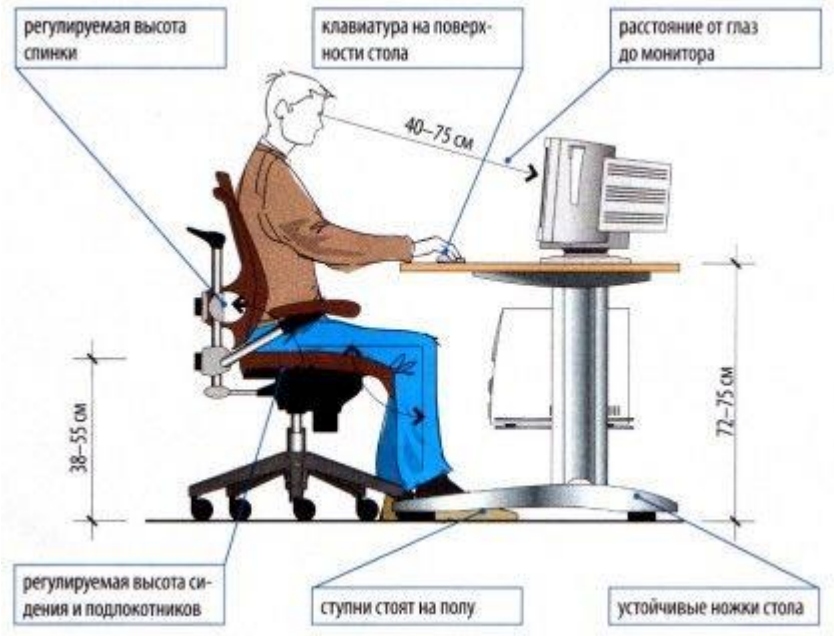

Рисунок 1 – Организация работы на ПЭВМ

Правильная поза и положение рук оператора являются весьма важными для исключения нарушений в опорно-двигательном аппарате и возникновению синдрома постоянных нагрузок.

Согласно СанПиНу 2.2.2.542-96 при 8-ми часовой рабочей смене на ВДТ и ПЭВМ перерывы в работе должны составлять от 15-20 минут каждые 2 часа работы.

ЗАКЛЮЧЕНИЕ ПО РАЗДЕЛУ

Проанализированы факторы рабочей зоны на предмет выявления их вредных проявлений, это микроклимат, недостаточная освещенность, повышенный шум и умственное перенапряжение. Были выявлены предполагаемые источники загрязнения окружающей среды, возникающие в результате предлагаемого проекта. Обозначены организационные мероприятия обеспечения безопасности, описаны основные источники чрезвычайных опасностей.

Таким образом, по результатам проведенных исследований в рамках раздела «Социальная ответственность» было установлено, что обеспеченными условиями труда на рабочем месте предупреждены и минимизированы риски воздействия вредных и опасных факторов производства. Рассмотрены меры, позволяющие такие условия обеспечить.

ЗАКЛЮЧЕНИЕ

В соответствии с поставленной целью — разработка алгоритмов глубокого обучения для анализа тональности текста и сравнение их эффективности с другими классификаторами на основе алгоритмов машинного обучения — были реализованы архитектуры свёрточной нейронной сети, рекуррентной нейронной сети с LSTM-блоками, а также проведено сравнение показателей качества их классификации с другими классификаторами.

При использовании модели мешка слов точность различных методов была значительна выше случайной (около 70%), однако применяя модель Word2Vec, удалось значительно улучшить точность работы алгоритмов (на несколько единиц). Однако нейронные сети показали лучшие результаты.

Точность классификатора на основе свёрточной нейронной сети оказалась 79.9%. Самую высокую точность показал классификатор на основе рекуррентной сети с LSTM-блоками — 83.3%.

Результаты исследования показывают, что использование глубоких нейронных сетей значительно улучшает точность анализа тональности текста. Преимущество рекуррентной сети на основе LSTM над свёрточной нейронной сетью в области анализа тональности уже было доказано в различных исследованиях, однако важно отметить, что в данной работе были реализованы простейшие архитектуры глубоких нейронных сетей. Улучшение параметров модели, использование более расширенной модели векторного представления слов Word2Vec, применение attention-механизмов позволит значительно увеличить эффективность бинарного классификатора для анализа тональности на основе глубоких нейронных сетей.

Возможным направлением для дальнейшей работы может быть распознавания ключевых слов, вносящих наибольший вклад в положительный или отрицательный отзыв. Введение модуля в эксплуатацию.

СПИСОК ЛИТЕРАТУРЫ

1. K. Tran et al. Evaluation of deep learning toolkits. https://github.com/zer0n/deepframeworks/blob/master/README.md. (дата обращения 25.05.2018).

2. Данные рецензий, используемых в работе, sentence polarity dataset v1.0. http://www.cs.cornell.edu/people/pabo/movie-review-data/. (дата обращения 25.05.2018).

3. Видяев, И.Г. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова, Н.В. Шаповалова, Л.Р. Тухватулина З.В. Криницына; Томский политехнический университет. − Томск: Изд-во Томского политехнического университета, 2014. – 36 с.

4. ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация. - М.: Издательство стандартов, 2001. – 4 с.

5. СанПиН 2.2.4/2.1.8.055-96 Электромагнитные излучения радиочастотного диапазона. Санитарные правила и нормы. - М.: Информационно-издательский центр Госкомсанэпиднадзора России, 1996. – 28 с.

6. ГОСТ Р 50571.17-2000 Выбор мер защиты в зависимости от внешних условий. – М.: Издательство стандартов, 2012. – 8 с.

7. ГОСТ 26522-85. Короткие замыкания в электроустановках. Термины и определения. Переиздание 2005. – М.: Стандартинформ, 2006.

8. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.-М.: Информационно-издательский центр Госкомсанэпиднадзора России, 2003. – 8 с.

9. СанПиН 2.1.8/2.2.4.1190-03. Гигиенические требования к размещению и эксплуатации средств сухопутной подвижной радиосвязи.-М.: Информационно-издательский центр Госкомсанэпиднадзора России, 2003. – 9 с.

10. T. Mikolov et al. Distributed representations of words and phrases and their compositionality. arXiv:1310.4546 [cs.CL], 2013.

11. T. Mikolov et al. Efficient estimation of word representations in vector space. arXiv:1301.3781 [cs.CL], 2013.

12. Google Research Team. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 [cs.DC], 2016.

13. M. Schrimpf. Should i use tensorflow? arXiv:1611.08903 [cs.LG], 2016.

14. К. К. Семёнов. Автоматическое дифференцирование функций, выраженное программным кодом. Вестник Саратовского государственноготехнического университета, 2011.

15. В. Д. Чабаненко. Модификации метода стохастического градиентного спуска для задач машинного обучения с большими объемами данных. Master’s thesis, Московский государственный университет имени М.В. Ломоносова, 2016.

16. J. Turian et al. Word representations: A simple and general method for semi-supervised learning. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, с. 384-394, 2010.

17. R. Kadlec et al. Improved deep learning baselines for ubuntu corpus dialogs. arXiv:1510.03753, 2015.

ПРИЛОЖЕНИЕ

Модель Word2Vec и её использование для рецензий

```
# kaggle
from bs4 import BeautifulSoup
import re
from nltk.corpus import stopwords
import nltk.data
import numpy as np
from gensim.models import word2vec
from methods import random_forest
import pandas as pd
from sklearn. model_selection import train_test_split
import gensim.models
import logging
logging. basicConfig (format='%(asctime)s : %(levelname)s : %(
message)s',
level=logging.INFO)
tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')
def get_word_vectors (data):
train , test = train_test_split (data , test_size =0.2 ,
random_state =42)
unlabeled_train = pd.read_csv(
"unlabeledTrainData.tsv", header =0, delimiter ="\t",
quoting =3)
sentences = []
for review in train['review']:
sentences += review_to_sentences (review , tokenizer )
```

```
for review in unlabeled_train ["review"]:
sentences += review_to_sentences (review , tokenizer )
model = word2vec.Word2Vec(sentences , workers =4, size =100 ,
min_count =40, window =10, sample =1e−3)
model.wv. save_word2vec_format (’150_features’)
model. init_sims (replace=True)
clean_train_reviews = []
for review in train[’review’]:
clean_train_reviews .append( review_to_words (review ,
remove_stopwords =True))
trainDataVecs = getAvgFeatureVecs ( clean_train_reviews , model ,
num_features )
clean_test_reviews = []
for review in test[’review’]:
clean_test_reviews .append(
review_to_words (review , remove_stopwords =True))
testDataVecs = getAvgFeatureVecs ( clean_test_reviews , model ,
num_features )
return trainDataVecs , testDataVecs , train[’sentiment’], test[
’sentiment’]
def review_to_words (review , remove_stopwords =False):
review_text = BeautifulSoup (review).get_text ()
review_text = re.sub("[^a−zA−Z]", " ", review_text )
words = review_text .lower ().split ()
if remove_stopwords :
stops = set( stopwords .words("english"))
words = [w for w in words if not w in stops]
```

```
return words
def review_to_sentences (review , tokenizer , remove_stopwords =False
):
raw_sentences = tokenizer .tokenize(review.strip ())
sentences = []
for raw_sentence in raw_sentences :
if len( raw_sentence ) > 0:
sentences .append( review_to_words (raw_sentence ,
remove_stopwords ))
return sentences
def makeFeatureVec (words , model , num_features ):
featureVec = np.zeros (( num_features ,), dtype="float32")
nwords = 0.
index2word_set = set(model. index2word )
for word in words:
if word in index2word_set :
nwords = nwords + 1.
featureVec = np.add(featureVec , model[word ])
featureVec = np.divide(featureVec , nwords)
return featureVec
def getAvgFeatureVecs (reviews , model , num_features ):
reviewFeatureVecs = np.zeros ((len(reviews), num_features ),
dtype="float32")
for review in reviews:
reviewFeatureVecs [counter] = makeFeatureVec (review , model
, num_features )
return reviewFeatureVecs
```

```
if __name__ == "__main__":
main ()
```