МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

МОДЕЛЬ МАШИННОГО ОБУЧЕНИЯ ДЛЯ КЛАССИФИКАЦИИ УДК

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Тимофеева Владислава Алексеевича

Научный руководитель:
ст. преподаватель кафедры ИиП _____ Лапшева Е. Е.

подпись, дата

Зав. кафедрой:
к.ф.-м.н., доцент _____ М.В. Огнева

подпись, дата

Саратов 2021

ВВЕДЕНИЕ

Актуальность темы. В настоящее время область искусственного интеллекта становится все более и более популярной. Все большее число компаний-гигантов включает машинное обучение в свои коммерческие разработки, появляется все больше алгоритмов, которые решают конкретные практические задачи. К этой области можно отнести и направление обработки естественного языка. Во многом успех в сфере искусственного интеллекта связан с значительными достижениями в области компьютерной техники, а именно возросшей мощности процессоров и объема оперативной памяти, что позволяет проектировать модели машинного обучения с большим числом параметров, что в свою очередь положительно влияет на результаты этих моделей при решении конкретной задачи.

Цель бакалаврской работы – решение задачи классификации научных работ согласно УДК с использованием моделей машинного обучения.

Поставленная цель определила **следующие задачи**:

1. Провести анализ существующих решений
2. Реализовать модели машинного обучения
3. Проанализировать эффективность моделей и выбрать наилучшую
4. Исследовать возможность использования моделей в ансамбле.

Методологические основы модели машинного обучения для классификации УДК представлены в работах С. Паттанаяка, А. Жерона, Й. Голдберга, К. Бокка, Б. Бенгфорта, С. Костадинова.

Практическая значимость бакалаврской работы.

Созданный алгоритм может быть использован как интегрированный в web-платформу, разработанную в ходе дипломной работы Лапушкиным Дмитрием Алексеевичем, так и как самостоятельное приложение, позволяющее по ключевым словам работы сопоставить УДК документа. Это в свою очередь поможет ускорить процесс классификации документа по УДК.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 56 страниц, из них 43 страниц – основное содержание, включая 21 рисунок, список использованных источников информации – 21 наименование.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Машинное обучение для обработки естественного языка**»** посвящен основным определениям машинного обучения и описанию алгоритмов классификации текстов. Кроме того перечислены некоторые виды нейронных сетей, а также алгоритмы их обучения.

Машинное обучение — это раздел искусственного интеллекта, который позволяет системе самостоятельно обучаться и улучшаться за счет применения знаний, накопленных от решения множества сходных задач.

Существует 4 основных метода машинного обучения:

1. Обучение с учителем — самый распространенный метод, при котором модель учится сопоставлять входные данные с уже размеченными выходными. Основные задачи, в которых используется обучение с учителем — классификация, регрессия.
2. Обучение без учителя — метод машинного обучения, при котором программа самостоятельно учится находить взаимосвязи между входными данными. Данный метод в основном решает задачу кластеризации данных.
3. Обучение с частичным привлечением учителя — является одной из разновидностей обучения с учителем. Главное отличие в том, что не все данные, подаваемые на вход, имеют разметку, и задача алгоритма сводится к поиску взаимосвязей между данными, включая часть заранее известных выходных данных.

4. Обучение с подкреплением — основная идея этого метода заключается в том, что программа, называемая агентом, учится взаимодействовать с окружающей средой. За правильные действия алгоритму начисляется награда, а за неправильные штраф. Целью алгоритма становится максимизация величины награды за заданный промежуток времени.

Обработка естественного языка, или *NLP* (natural language processing), относится к отрасли информатики, а точнее, к отрасли искусственного интеллекта (ИИ), связанной с предоставлением компьютерам возможности понимать текст и произносимые слова так же, как это могут делать люди. Данный раздел включает алгоритмы машинного обучения, позволяющие обрабатывать как текст, так и речь пользователя.

Токенизация — это задача разбиения определенной последовательности символов на части, называемые токены, отбрасывая при этом некоторые ненужные слова, называемые стоп-словами. Простейшим примером токенизации может быть разбиение текста на предложения и отдельные слова.

Лемматизация — это последовательность действий, целью которой является нахождение начальной (словарной) формы слова в зависимости от его грамматического значения. Для различных частей речи лемма будет отличаться. Так для имени существительного и имени прилагательного леммой является форма именительного падежа и единственного числа (мужского рода для имени прилагательного), а для глаголов, причастий и деепричастий результатом лемматизации будет глагол несовершенного вида в неопределенной форме.

Стемминг — это последовательность действий, целью которой является нормализация слова, удаление из него окончаний, формообразующих суффиксов и т.п.

Стоп-слова — слова, которые в большом количестве присутствуют в тексте, однако не вносят в него никакой дополнительной информации, лишь добавляют шум в данные. Это могут быть различные предлоги, союзы и т.п.

Векторизация — процесс перевода текстовой информации в векторное пространство с использованием различного рода алгоритмов.

Алгоритм k-ближайших соседей является простейшим метрическим классификатором, основная идея которого заключается в оценке сходства между входным элементом и элементами обучающей выборки. В качестве меры сходства может выступать евклидово, манхэттенское или косинусное расстояние, которое обычно используется для сравнения слов в векторном пространстве. Несмотря на свою простоту, при определенных данных и правильно подобранном числе соседей этот алгоритм может показывать достаточно высокую точность. В итоге алгоритм классификации можно представить следующим образом: для входной точки вычисляется расстояние до всех точек в соответствии с заданной мерой расстояния; далее выбирается k-точек, расстояние до которых является наименьшим; результатом классификации является класс, наиболее часто встречающийся среди этих k-точек.

Дерево решений является одним из часто используемых алгоритмов машинного обучения для решения задачи классификации. Основная идея алгоритма заключается в том, что для каждого атрибута в наборе данных формируется узел, в котором наиболее важный атрибут помечается как корень. Алгоритм продолжается до тех пор, пока не будет достигнут конечный узел, содержащий прогноз или результат дерева решений.

Случайный лес (Random forest) — алгоритм машинного обучения, который заключается в использовании ансамбля решающих деревьев. Суть алгоритма заключается в использовании большого ансамбля решающих деревьев, каждое из которых хоть и дает невысокие показатели классификации, но за счет большого количества деревьев совокупная точность модели находится на высоком уровне. Для задач регрессии

результатом алгоритма будет усредненное значение выхода всех деревьев, для задачи классификации – значение, которое выдало наибольшее количество деревьев.

Искусственная нейронная сеть — это математическая модель, а также ее реализация на программном или аппаратном уровне. По своей структуре она имитирует биологическую нейронную сеть — сеть нервных клеток живого организма. Искусственная нейронная сеть состоит из слоев нейронов, которых, в зависимости от решаемой задачи, может быть более двух. Первый слой нейросети называется входным, последний — выходным; все слои, которые находятся между входным и выходным слоями называются скрытыми.

Сверточная нейронная сеть или *CNN* (convolutional neural network) - свое название получила из-за операции свертки, применяемой на ее слоях, по своей сути представляет матричное умножение. Слой свертки включает в себя особый фильтр для каждого канала, ядро свертки которого обрабатывает предыдущий слой по фрагментам (суммируя результаты матричного произведения для каждого фрагмента). Весовые коэффициенты ядра свертки неизвестны и устанавливаются в процессе обучения. Ядро свертки обычно имеет небольшую размерность, как правило это 3×3, 5×5, 7×7 и 9×9, однако допускаются и другие размерности.

Рекуррентная нейронная сеть или *RNN* (recurrent neural network) — это тип искусственной нейронной сети, которая использует последовательные данные или данные временных рядов. Эти алгоритмы глубокого обучения обычно используются для порядковых или временных задач, таких как языковой перевод, обработка естественного языка (NLP), распознавание речи и добавление субтитров к изображениям. Как и нейронные сети с прямой связью и сверточные нейронные сети (CNN), рекуррентные нейронные сети используют обучающие данные для обучения. Они отличаются своей «памятью», поскольку берут информацию из предыдущих входов, чтобы влиять на текущий ввод и вывод. В то время как

традиционные глубокие нейронные сети предполагают, что входы и выходы независимы друг от друга, выходные данные рекуррентных нейронных сетей зависят от предшествующих элементов в последовательности.

Рекуррентные нейронные сети используют алгоритм обратного распространения во времени или BPTT (backpropagation through time) для определения градиентов, который немного отличается от традиционного обратного распространения, поскольку он специфичен для данных последовательности. Принципы BPTT такие же, как и при традиционном обратном распространении, где модель обучается, вычисляя ошибки от выходного уровня до входного. Эти расчеты позволяют соответствующим образом скорректировать и подогнать параметры модели. BPTT отличается от традиционного подхода тем, что суммирует ошибки на каждом временном шаге, тогда как в сетях прямого распространения нет необходимости суммировать ошибки, поскольку они не разделяют параметры на каждом уровне.

Двунаправленные рекуррентные нейронные сети (BRNN): это вариант сетевой архитектуры RNN. В то время как однонаправленные RNN могут извлекаться только из предыдущих входных данных для прогнозирования текущего состояния, двунаправленные RNN извлекают будущие данные для повышения их точности. Например, чтобы предсказать следующее слово в предложении, часто полезно знать контекст вокруг слова, а не только слова, идущие перед ним.

Сети с долговременной краткосрочной памятью или LSTM (long short-term memory) представляют собой модифицированную версию рекуррентных нейронных сетей, которая упрощает запоминание прошлых данных в памяти. Здесь решается проблема исчезающего градиента RNN. LSTM хорошо подходит для классификации, обработки и прогнозирования временных рядов с учетом временных лагов неизвестной длительности. Она обучает модель с помощью обратного распространения. В сети LSTM присутствует три шлюза:

1. Input gate — определяет, какие входные значения должны быть использованы для изменения памяти.
2. Forget gate — определяет какие значения должны быть «забыты» на данном шаге.
3. Output gate — агрегирует в себе результаты, находящиеся в памяти и поступающие на вход.

Управляемые рекуррентные блоки или GRU (gated recurrent unit): этот вариант RNN похож на LSTM, поскольку он также работает для решения проблемы краткосрочной памяти моделей RNN. Вместо использования регулирующей информации о «состоянии ячейки» он использует скрытые состояния, и вместо трех вентилей у него есть два — вентиль сброса и вентиль обновления. Подобно воротам в LSTM, шлюзы сброса и обновления контролируют, сколько и какую информацию сохранить.

Градиентный спуск является одним из многих алгоритмов машинного обучения. Суть этого метода заключается в нахождении экстремума (минимума) функции, в случае нейросетей — это функция потерь (loss function).

Ансамблевые методы — это парадигма машинного обучения, в которой несколько моделей обучаются для решения одной и той же проблемы и объединяются для повышения результативности.

Второй раздел «Универсальная десятичная классификация» посвящен определению понятия универсальной десятичной классификации.

Универсальная десятичная классификация (УДК) — это система, используемая для кодировки публикуемых текстов, основанная на иерархическом принципе от общего к частному и учитывающая тематику и исследовательское направление работы. При этом применяется цифровой десятичный код. Такая система кодирования позволяет обнаружить в любой библиотеке или электронном хранилище нужную публикацию, сборники научных статей, не прибегая к длительным поискам.

Третий раздел «Реализация модели машинного обучения для получения УДК текста» посвящен описанию процесса реализации алгоритма классификации текста согласно УДК.

Всю реализацию алгоритма можно разделить на 4 этапа:

1. Предобработка данных — включает в себя нормализацию распределения классов УДК внутри обучающей выборки, которая была получена в ходе дипломной работы Лапушкина Дмитрия Алексеевича. В качестве функции нормализации была выбрана медианная мера.
2. Реализация и тестирование стандартных алгоритмов классификации. В ходе этого этапа были использованы встроенные в пакет sklearn алгоритмы k-ближайших соседей, дерево решений и случайный лес. Для каждого из них итеративным путем были подобраны параметры влияющие на результирующую точность. Результат работы алгоритмов можно увидеть в таблице 1.
3. Реализация и тестирование моделей нейронный сетей. Так как стандартные алгоритмы не показали достаточной для решения задачи точности, было решено использовать модели нейронных сетей. В итоге были спроектированы модель сверточной нейронной сети и де рекуррентных с разным типом ячеек (ячейки долгосрочной памяти и управляемой). Наилучший результат показала рекуррентная нейронная сеть с ячейками долгосрочной памяти. Результат тестирования можно увидеть в таблице 2.
4. Исследование работы нейронных сетей в ансамбле. Последним этапом реализации алгоритма стала работа вышеупомянутых нейронных сетей в ансамбле. Ансамбль нейронных сетей работал по следующему принципу: входные ключевые слова подавались на вход каждой модели независимо; затем результат классификации усредняется и выбирался наивысший показатель среди этого усреднения. Данное решение позволило повысить точность работы алгоритма до 0.84.

Алгоритм	Точность
k-ближайших соседей	0.5190
Дерево решений	0.6123
Случайный лес	0.6320

Таблица 1 - тестирование стандартных алгоритмов классификации

Модель нейронной сети	Точность
Сверточная нейронная сеть (CNN)	0.7891
Рекуррентная с ячейками долгосрочной памяти	0.8110
Рекуррентная с ячейками управляемой памяти	0.8011

Таблица 2 - тестирование моделей нейронных сетей

ЗАКЛЮЧЕНИЕ

Целью данной бакалаврской работы являлось решение задачи классификации научных работ согласно УДК с использованием модели машинного обучения. Для достижения данной цели было проанализировано множество англоязычных и русскоязычных источников литературы по таким темам, как обработка текстовой информации и построение нейронных сетей. В результате были созданы и протестированы модели нейронных сетей, а именно сверточная нейронная сеть, рекуррентные нейронные сети с ячейками долгосрочной и управляемой памяти, а также протестирована работа данных сетей в ансамбле. В итоге для конечной реализации из вышеописанных нейронных сетей был выбран ансамбль, который показал максимальную точность на тестовой выборке, а именно 84 процента.

Получившаяся модель была интегрирована в web-платформу, которая была разработана Лапушкиным Дмитрием Алексеевичем в ходе выполнения своей выпускной квалификационной работы. В итоге механизм работы платформы сводится к следующему: на сайт загружается документ, из

которого извлекаются ключевые слова и подаются на вход модели нейронной сети, которая затем возвращает УДК работы и вероятность этого УДК-класса обратно веб-сервису.

Деятельность, проведенная в рамках данной дипломной работы не полностью решает поставленную задачу, так как количество классов ограничено в силу объема собранных данных. Данная проблема может быть решена путем покупки API ресурсов, благодаря которым можно получить больше данных для обучения, вследствие чего может быть увеличено количество классов и точность существующей модели.

Основные источники информации:

1. Паттанаяк, С. Глубокое обучение и TensorFlow для профессионалов/С. Паттанаяк; пер. А.Г. Гузикевич, ред. В.Р. Гинзбург — М.: Вильямс, 2019. — 480 с.
2. Géron, A. Hands-on Machine Learning with Scikit-Learn and TensorFlow/A.Géron//Sebastopol: O'Reilly, 2018. — 856 p.
3. Bengfort, B. Applied Text Analysis with Python/B. Bengfort// Sebastopol: O'Reilly, 2018. — 332 p.
4. Goldberg, Y. Neural Network Methods in Natural Language Processing/Y. Goldberg//Ramat Gan: Bar Ilan University, 2017. — 310 p.
5. Bokka, K.R. Deep Learning for Natural Language Processing/K.R. Bokka//Birmingham: Packt, 2019. — 372 p.
6. Kostadinov, S. Recurrent Neural Networks with Python Quick Start Guide: Sequential learning and language modeling with TensorFlow/S. Kostadinov//Birmingham: Packt, 2018. — 122 p.