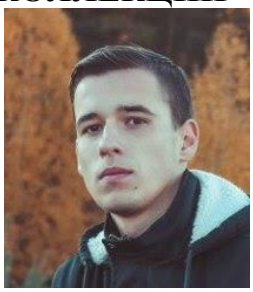
УДК 004.032.26

СИСТЕМА АНАЛИЗА КАЧЕСТВА ТЕКСТОВЫХ КОЛЛЕКЦИЙ

А. Л. Калоша
Магистрант БГУИР, инженер-программист JazzTeam

М.А. Медунецкий
Студент БГУИР



М.П. Хоронеко
Студент БГУИР

А.А. Александров
Студент БГУИР

А.И. Гридасов
Старший преподаватель каф информатики БГУИР

С.Н. Нестеренков
Кандидат технических наук, доцент БГУИР

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
JazzTeam, Республика Беларусь
E-mail: andreikalosha@mail.ru

А. Л. Калоша

Окончил Барановичский государственный университет. Магистрант БГУИР. Сейчас работает в должности инженера-программиста. Занимается разработкой системы предиктивного анализа для классификации документов текстовых коллекций.

А.И. Гридасов

Окончил БПИ (ныне БНТУ) в 1985 г, сейчас работает старшим преподавателем БГПУ им. М. Танка на кафедрах физики и методики преподавания физики, информатики и методики преподавания информатики.

С.Н. Нестеренков

Окончил БГУИР в 2013 г, сейчас работает на кафедре Программного обеспечения информационных технологий. Занимается научными исследованиями по направлениям: "Модели и методы искусственного интеллекта", "Информационные системы и технологии", "Современные технологии управления разработкой программного обеспечения".

Аннотация. Цель данной работы заключается в создании системы для прогнозирования популярности публикаций. В данной системе используется нейронная сеть, которая обучена на наборе метрик, описывающих качество и популярность публикаций. В качестве набора метрик используется количество лайков, просмотров и репостов. Обучение нейронной сети производилось на 100 000 текстов. В результате обучения нейронная сеть способна предсказать количество просмотров с точностью в 75%. Верным считается ответ, находящийся в диапазоне +/-200 000 просмотров от ответа. Максимальное количество просмотров при обучении составляло 48 миллионов. Коэффициент корреляции для массивов ответов и предсказанных значений составляет 0,33. Это означает, что между входными и выходными данными есть линейная зависимость. Увеличив размер обучающей выборки, или подобрав более точно гиперпараметры нейронной сети, можно увеличить точность системы.

Ключевые слова: Big Data аналитика, TensorFlow, CUDA, машинное обучение, нейронные сети.

Введение. Объем информации, доступной в сети Интернет, растет с каждым годом. Причем большая часть этой информации представляет собой тексты на естественном языке. В зависимости от области знаний, информация может быть представлена в виде статей, комментариев или сообщений на публичном форуме. Информация в сети Интернет дублируется, уточняется и пополняется ежедневно. Нетрудно понять, что имеющиеся в данный момент доступные ресурсы всемирной сети представляют собой колоссальную базу знаний, представленных в форме, сложно поддающейся компьютерной обработке – в виде текста [1].

Как правило, изучить весь контент (текст) не представляется возможным даже в отдельных областях, поэтому приходится фильтровать получаемую информацию и выбирать лучшую.

Назначение разрабатываемой системы заключается в предсказании популярности статей через определенный промежуток времени. Статья считается популярной при высоком количестве лайков, репостов или просмотров. Данные метрики зависят от множества факторов, таких как название, авторов, время публикации и содержание статьи. Эти параметры наилучшим образом отражают популярность (качество) статьи. Правильно обученная нейронная сеть позволяет с высокой точностью предсказать значения метрик популярности неопубликованного контента.

Для обучения нейронной сети была выбрана библиотека TensorFlow как один из лучших инструментов машинного обучения. TensorFlow — это библиотека программного обеспечения с открытым исходным кодом для численного расчета с использованием графиков потока данных [2].

Нейронная сеть — это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющих их для последующей обработки [3].

Нейронная сеть сходна с мозгом с двух точек зрения:

Знания поступают в нейронную сеть из окружающей среды и используются в процессе обучения;

Для накопления знаний применяются связи между нейронами, называемые синоптическими весами [3].

Архитектура нейронной сети. На рисунке 1 проиллюстрирована архитектура нейронной сети, которая состоит из 4 слоев (входной, два промежуточных и выходной слой).

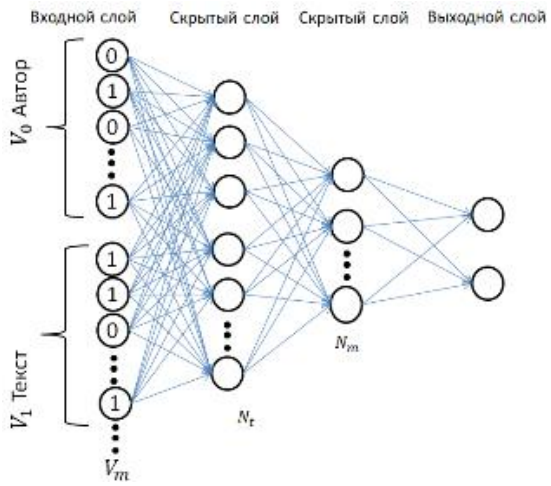


Рисунок *1*. – Архитектура нейронной сети

На промежуточных слоях используется функция активации LeakyReLU, на выходном слое применяется функция softmax. Между всеми слоями, кроме последнего, используется нормализация данных. Одной из главных проблем при обучении нейронной сети является переобучение. Когда нейронная сеть перестает обучаться, а просто запоминает ответы из обучающей выборки, для минимизации ошибки, тем самым уходя от первичной цели. Для решения этой проблемы использовалась регуляризация и dropout.

Обучение нейронной сети. Для обучения нейронной сети необходимо большое количество статей и метаданных, таких как автор, дата создания, ключевые слова и другие.

Перед обучением данные делятся на 2 части: для тестирования и для обучения.

Опишем процедуру обучения нейронной сети. На вход нейронной сети подается матрица векторов MV, каждый вектор V которой содержит информацию о конкретном атрибуте публикации (например, авторе). Для формирования отдельного вектора V перед обучением необходимо получить словарь D всех значений атрибута публикации. Словарь D сортируется по убыванию и отбрасываются последние N значений, чтобы нейронная сеть не обучалась на редко встречающихся элементах, и тем самым не ухудшалась точность классификации. Указанная выше процедура выполняется для каждого атрибута. Для каждого автора публикации, производится поиск в словаре D, если данный автор найден, то под индексом найденного автора в вектор V ставится единица, иначе – ноль. Таким образом, заполняются все векторы матрицы MV [4].

Выходной вектор R описывает количество просмотров через заданный промежуток времени и состоит из единственного дробного числа, находящегося в диапазоне от ноля до единицы. Единица означает максимальное количество просмотров, в данном исследовании выбрано 50 миллионов [4].

Промежуток времени, на который нейронная сеть способна предсказать популярность публикации, является статическим и определяется до обучения нейронной сети. Т.е чтобы изменить этот параметр нужно обучить нейронную сеть заново. Для предсказания популярности публикации через несколько временных отрезков, например, неделя, месяц и год можно использовать два варианта:

–Обучить несколько нейронных сетей;

–Изменить архитектуру нейронной сети таким образом, чтобы на выходном слое был вектор, содержащий значения популярности для нескольких временных интервалов.

У каждого из способов есть достоинства и недостатки, и выбирать нужно, исходя из постановки задачи.

Плюсом при использовании первого варианта, является проста реализации и тестирования приложения. Минусом является необходимость поддержания нескольких копий приложения, по одному на каждый из временного интервала.

Плюсом при использовании второго варианта является необходимость поддержания только одного экземпляра приложения вместо нескольких, как в первом варианте.

Минусом является сложность создания архитектуры, создания приложения и оценки результата, т.к. нейронная сеть может обучиться предсказывать некоторые временные участки лучше других, хотя в среднем результат будет оптимальным.

Существует прямая зависимость между скоростью обучения нейронной сети и точностью предсказания. Для ускорения процесса обучения используется вычислительная мощность видеокарты, а именно технология CUDA. CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU (графических процессоров) [5].

Обучение нейронной сети на CPU занимает от 4 до 16 часов, в зависимости от глубины обучения и точности результата. В то время обучения на GPU не превышает часа.

Гиперпараметры — это значения, которые нужно подбирать вручную и зачастую методом проб и ошибок. Среди таких значений можно выделить:

- Скорость обучения;
- Количество скрытых слоев;
- Количество нейронов в каждом слое;
- Параметры для нормализации, регуляризации и dropout.

Выбор правильного количества нейронов в скрытых слоях является очень важным. Слишком малое количество – и сеть не сможет обучиться. Слишком большое повлечет за собой увеличение времени обучения сети до фактически нереального значения. Также это может привести к переобученности сети (overfitting), проявляющейся в том, что сеть будет прекрасно работать на обучающей выборке, но очень плохо на входных примерах не входящих в нее [6].

Это происходит из-за того, что сеть будет обладать избыточными способностями к обучению, и наряду со значительными для данной задачи факторами будет учитывать черты, характерные лишь для данной обучающей выборки [6].

Однако, существуют эвристические правила выбора количества нейронов в скрытых слоях. Одним из таких правил является правило геометрической пирамиды (geometric pyramid rule). По этому правилу число нейронов скрытого слоя в 4-exслойном перцептроне вычисляемая по следующим формулам:

$$r = \sqrt[3]{\frac{n}{m}}$$
$$k_1 = mr^2$$
$$k_2 = mr$$

где k1 – число нейронов в первом скрытом слое; k2 – число нейронов во втором скрытом слое [6].

Хотя существует еще более точный, но и более долгий способ. Он состоит в том, что сначала используется сеть с одним скрытым слоем с одним, двумя нейронами. Если она смогла достигнуть необходимого уровня ошибки, то процесс обучения закончен, иначе добавляем еще один нейрон и так до тех пор, пока ошибка сети не станет приемлемо малой, или до тех пор, пока увеличение числа нейронов не сможет значительно улучшить характеристики сети [6].

Экспериментальным путем было установлено, что два скрытых слоя является оптимальным выбором для решения данной задачи.

Оптимизатор — это алгоритм, который изменяет веса и смещения во время обучения [7]. В данной задаче был выбран оптимизатор "adam", имеющий высокую скорость и точностью решения.

Adam – это алгоритм оптимизации, который можно использовать вместо классической процедуры стохастического градиентного спуска для итеративного обновления весов сети на основе обучающих данных. Плюсом данного метода является высокая производительность в случае решения задач оптимизации на больших наборах данных, что также является причиной выбора данного оптимизатора [7].

TensorBoard – это интегрированная среда визуализации графика TensorFlow и анализа записанных метрик во время обучения и вывода. Данный инструмент использовался для контроля качества обучения нейронной сети. Это особенно удобно при подборе оптимальных гиперпараметров для обучения нейронной сети [8].

В качестве метрик качества выступает точность (accuracy). Точность(accuracy) – это доля правильных ответов. Верным считается ответ, находящийся в диапазоне +/-200 000 просмотров от ответа.

Функция потерь – функция, которая в теории статистических решений характеризует потери при неправильном принятии решений на основе наблюдаемых данных [9]. В качестве функции потерь используем среднеквадратическую ошибку (Mean Squared Error). Данная функция потерь была выбранная для данной задачи из-за высокого штрафа за значительное отклонение предсказанного значения от правильного ответа [10]. Значение функции ошибки также используется для оптимизации гиперпараметров нейронной сети. TensorBoard позволяет отобразить график функции ошибки для тестовой и обучающей выборки на каждом шаге обучения (Рисунок 2).

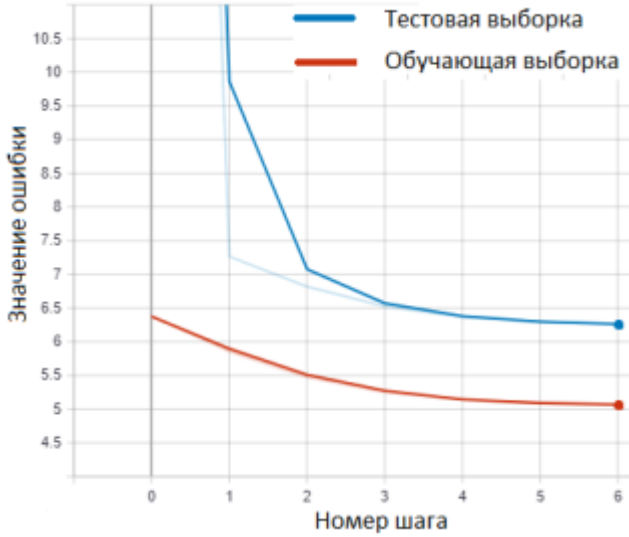


Рисунок 2. – График функции ошибки для тестовой и обучающей выборки

Исходя из графика, можно сделать вывод, что для увеличения точности необходимо увеличить обучающую выборку, или подобрать гиперпараметры таким образом, чтобы значение ошибки на тестовой и валидационной выборках не отличалось. Так же значение ошибки достаточно велико что может свидетельствовать о недостатке данных для принятия решения. То есть нужно добавить новые атрибуты для обучения.

Увеличение размера обучающей выборки или добавление новых атрибутов увеличивает затраты оперативной памяти. В данной работе при обучении нейронной сети нередко достигался предел оперативнйо памяти в 8 ГБ. Для преодоления данного предела нужно использовать методы снижения размерности, либо использовать оборудование с большим запасом оперативной памяти.

Тестирование. Обучение и тестирование нейронной сети было произведено на компьютере со следующими характеристиками:

- процессор Intel Core i5-8600;
- оперативная память 8,0 ГБ;
- видеокарта Nvidia GeForce GTX 2060;
- частота графического процессора 1365 МГц;
- 6144 Мб видеопамяти GDDR6;
- частота видеопамяти 3500 МГц;
- 1920 cuda ядер.

Для тестирования была выбрана выборка размером 25000 текстов не участвующих в обучении.

Для каждого из тестовых текстов нейронная сеть делает предсказание и записывает результат в excel. Также в данный файл записываются правильные ответы. Верным считается ответ, находящийся в диапазоне +/-200000 просмотров от верного ответа. Соблюдение данного условия считается при помощи формул в excel для каждого текста. В результате нейронная сеть способна предсказать количество просмотров с точностью в 75%.

Для оптимизации гиперпараметров также производился анализ результатов нейронной сети на основе сравнения графиков верных и предсказанных значений (Рисунок 3).

Рисунок 3. – Результат предсказания популярности 17 публикаций

Как видно из графика, нейронная сеть достаточно точно предсказывает популярность публикаций. Для улучшения результатов нужно увеличить обучающую выборку, а также подобрать более точные значения гиперпараметров.

Заключение. Обучение нейронной сети производилось на 100 000 текстов. В результате обучения нейронная сеть способна предсказать количество просмотров с точностью в 75%. Верным считается ответ, находящийся в диапазоне +/-200 000 просмотров от ответа. Максимальное количество просмотров при обучении составляло 48 миллионов. Коэффициент корреляции для массивов ответов и предсказанных значений составляет 0,33. Это означает, что между входными и выходными данными есть линейная зависимость. Увеличив размер обучающей выборки, или подобрав более точно гиперпараметры нейронной сети, можно увеличить точность системы [4].

Список литературы

[1.] Степанов, П.А. Системы анализа текстов естественного языка / П.А. Степанов. – Тамбов: Грамота, 2013. – С. 159-161.

[2.] Library for numerical computation using data flow graphs [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/. – Дата доступа: 15.02.2020г.

[3.] Хайкин, С. Нейронные сети: полный курс, 2-е издание/ С. Хайкин. — М. : Издательский дом «Вильямс», 2006. — 1104 с.

[4.] Калоша, А. Л. Система предиктивного анализа для классификации документов текстовых коллекций / А. Л. Калоша, М. А. Медунецкий, М. П. Хоронеко // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018 / editorial board: M. Batura [etc.]. – Minsk, BSUIR, 2018. – P. 467 – 468.

[5.] CUDA parallel computing [Электронный ресурс]. – Режим доступа: http://www.nvidia.ru/object/cuda-parallel-computing-ru.html – Дата доступа: 15.02.2020г.

[6.] Выбор параметров нейронной сети [Электронный ресурс]. – Режим доступа: http://mei06.narod.ru/sem7/iis/shpora/page2_9.html – Дата доступа: 15.02.2020г.

[7.] Алгоритм оптимизации Адам для глубокого обучения [Электронный ресурс]. – Режим доступа: https://www.machinelearningmastery.ru/adam-optimization-algorithm-for-deep-learning – Дата доступа: 15.02.2020г.

[8]. Шакла, Е. Машинное обучение и TensorFlow / Н. Шакла. – СПб: Питер, 2018. – С. 336.

[9.] Klebanov, L. Robust and Non-Robust Models in Statistics / L. Klebanov, S.T. Rachev, F. Fabozzi. – New York: Nova Scientific Publishers, 2009. P. 305.

[10.] Dekking, M. A modern introduction to probability and statistics: understanding why and how / M. Dekking. – London: Springer, 1946. P. 318.

TEXT COLLECTION QUALITY ASSURANCE SYSTEM

A.L. Kalosha
Master student of the BSUIR, software engineer JazzTeam

M.A. Meduneckij
Student of the BSUIR

M.P. Horoneko
Student of the BSUIR

A.A. Aleksandrov
Student of the BSUIR

A.I. Gridasov
Senior Lecturer, Department of Informatics, BSUIR

S.N. Nesterenkov
Candidate of Technical Sciences, Associate Professor, BSUIR

Belarussian State University Informatics and Radioelectronics, Republic of Belarus
JazzTeam, Republic of Belarus
E-mail: andreikalosha@mail.ru

Abstract. The purpose of this work is to create a system for forecasting the popularity of publications. This system uses a neural network, which is trained on a set of metrics describing the quality and popularity of publications. The number of likes, views and repostings is used as a set of metrics. The neural network was trained for 100,000 texts. As a result of training the neural network is able to predict the number of views with the accuracy of 75%. The answer in the range of +/-200 000 views of the answer is considered correct. The maximum number of views during training was 48 million. The correlation coefficient for answer arrays and predicted values is 0.33. This means that there is a linear relationship between input and output data. By increasing the size of the learning sample, or by more accurately selecting the hyperparameters of the neural network, you can increase accuracy of the system.

Keywords: Big Data analytics, TensorFlow, CUDA, machine learning, neural networks.