МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Анализ текста с помощью нейронных сетей

Курсовой проект

Соболевского Дениса
Сергеевича
студента 3 курса, 4 группы
специальность
“Информатика”
Научный рукводитель:
Старший преподаватель кафедры КТС
Шолтанюк С. В.

Минск, 2020

АННОТАЦИЯ

Соболевский Д. С. Анализ текста с помощью нейронных сетей: Курсовой проект / Минск БГУ, 2020. - 20 стр.

В данной работе рассматриваются принципы работы нейронных сетей и их приложения для анализа текста, проводится анализ результатов обучения на примере.

АНАТАЦЫЯ

Сабалеўскі Д. С. Аналіз тэксту з дапамогай нейронных сетак: Курсавы праект / Менск: БДУ, 2020 - 20 стар.

У працы разглядаются прынцыпы працы нейронных сетак і іх прымяненні ў аналізе тэксту, праводзіцца аналіз вынікаў абучэння на прыкладзе.

ANNOTATION

Sobolevsky D.S. Neural networks in text analysis problems: Course project / Minsk: BSU, 2020 - 20 p.

In this paper, basic neural networks working principles and its applications for text analysis are discussed. The paper also includes an example problem solution and its results review.

ВВЕДЕНИЕ

Изначально, научный термин “нейронная сеть” появился в середине 20 века. Первые работы по данному направлению были представлены в 1943 году Мак-Каллоком и Питтсом. Основной идея заключалась в том, что нейрон можно представить как компьютерную модель, оперирующую двоичными числами, и что эти компьютерные модели способны обучаться путем подгонки параметров, описывающих синаптические связи биологических нейронов в мозге.

Такое направление получило большой интерес и развитие со стороны научного сообщества, однако в 1969 году вышла работа Минского и Пейперта, показывающая основные проблемы нейронных сетей. Первая заключалась в том, что однослойные нейросети не могли совершать операцию “сложение по модулю 2”. Второй важной проблемой было то, что компьютеры того времени не обладали достаточной вычислительной мощностью, чтобы обучать даже небольшие нейронные сети. После публикации этой работы интерес к нейросетям заметно угас, но снова возродился уже в середине 80-ых годов в связи с появлением алгоритма обратного распространения ошибки, а после того, как вычислительная мощность компьютеров стала расти в геометрической прогрессии, теория нейронных сетей получила еще больший импульс для исследований.

На сегодняшний день нейронные сети имеют огромное множество применений, основные из которых - задачи распознавания образов и классификации, принятие решений в зависимости от ситуации, задачи кластеризации, прогнозирования, аппроксимации, оптимизации.

Стоит отметить, что нейронные сети получили большое применение как и в некоммерческих сферах жизни (например, медицина, гражданская авиация), так и в коммерческих продуктах многих компаний, что свидетельствует о том, что нейронные сети достаточно распространены в современном мире.

Целью данной курсовой работы является изучение особенностей работы нейронных сетей для задачи анализа текста, а также программная реализация решения задачи классификации спама в смс-сообщениях и анализ полученных результатов.

1. Основные понятия теории нейронных сетей

1.1 Структура нейронной сети. Понятие искусственного нейрона

Нейросеть представляет собой набор взаимодействующих компонент - нейронов (искусственных нейронов). Каждый нейрон взаимодействует только с сигналом, который подается ему на вход, и с сигналом, который он передает на выход. Сам по себе искусственный нейрон - достаточно простая вычислительная структура, однако, образуя набор из большого и упорядоченного количества таких структур, они способны выполнять задачи сравнительно высокой сложности.

Рассмотрим структуру искусственного нейрона.

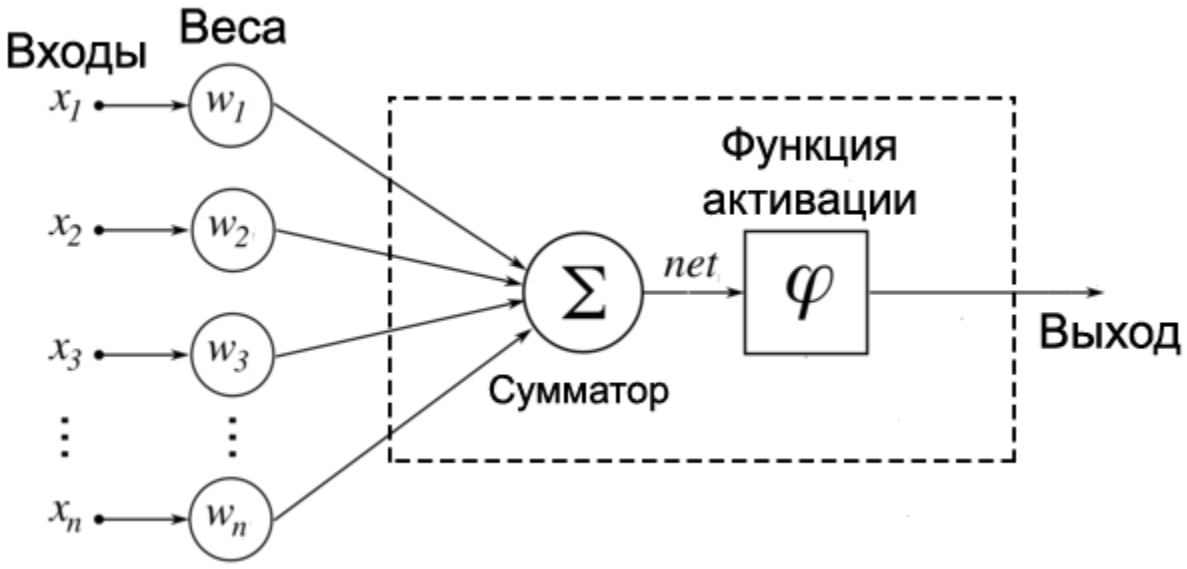

Рисунок 1. Структура искусственного нейрона

Входами нейрона могут быть либо входные данные поставленной задачи, либо выходы других, уже существующих, нейронов. Каждое входное значение x_i имеет свой вес w_i. Состояние нейрона определяется формулой:

$$S = \sum_{i=1}^{n} x_i w_i$$

Выходное значение нейрона определяется по формуле:

$$Y = \varphi(S),$$

где φ — функция активации.
Смысл применения функции активации будет рассмотрен в следующей главе.

1.2 Понятие функции активации и принципы ее выбора

Идея функции активации состоит в том, чтобы она выдавала либо 1 (“Да”), либо 0 (“Нет”) в зависимости от значения входных данных.
Допускается использование совершенно разных функций активации, (как правило, функция зависит от поставленной для нейросети задачи), однако она должна соответствовать следующим принципам:

1. Функция активации должна быть дифференцируема (т.к. для корректировки весов применяется алгоритмы градиентного спуска)
2. Функция активации не должна быть линейна (если она будет таковой, то в случае построения многослойной сети значения последнего слоя будут зависеть лишь от самого первого слоя)

Одной из самых распространенных функций активации является логистическая функция или сигмоида:

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

Сигмоида дифференцируема на всей оси абсцисс, что является необходимым условием выбора функции активации, как было отмечено выше, и, более того, сигмоида обладает свойством усиливать слабые сигналы лучше, чем сильные, а также предотвращает насыщение от больших сигналов так как они соответствуют областям аргументов, где сигмоида имеет пологий наклон.

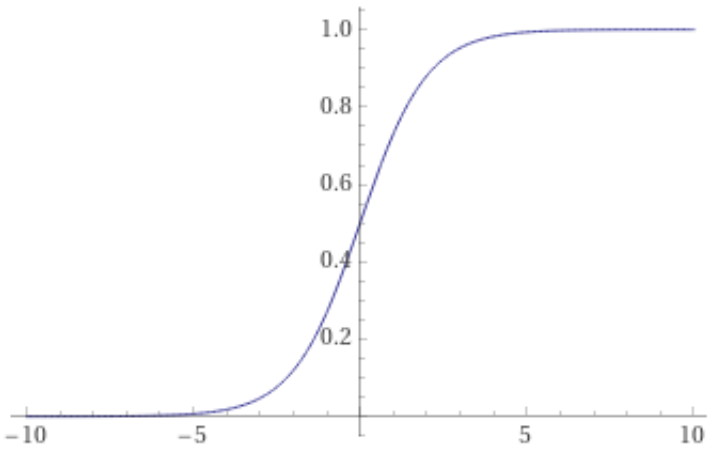

Рисунок 2. График сигмоиды

1.3 Многослойная нейронная сеть.

Выделяют также такое понятие, как слой нейронов - это совокупность нейронов, на который подается один и тот же сигнал.

Однако, нейросети редко состоят из одного нейрона и из одного слоя нейронов. Наиболее частая архитектура нейросетей - многослойная, как на примере ниже.

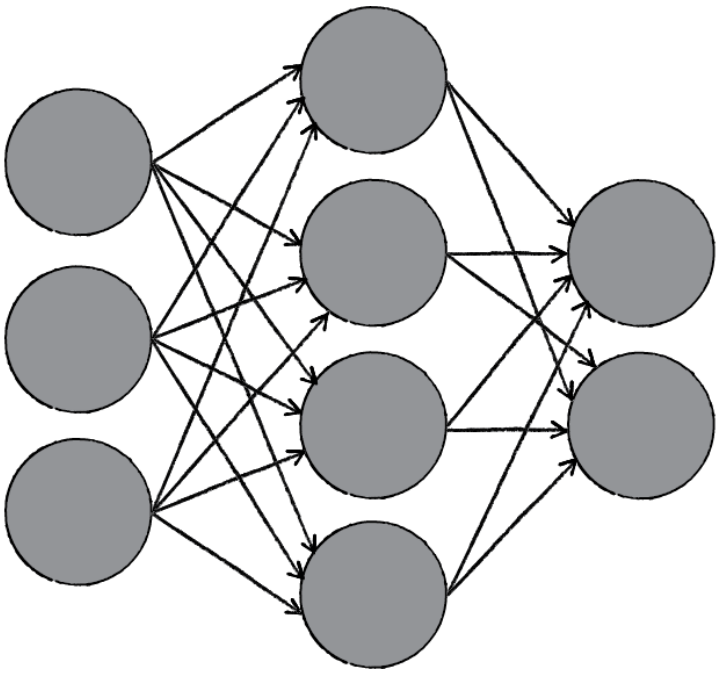

Рисунок 3. Трехслойная нейронная сеть

В случае многослойной нейронной сети выходные значения слоя нейронов подаются на входные узлы нейронов следующего строя.

1.4 Обучение нейронных сетей

Обучение нейронных сетей - это процесс, в котором параметры нейронной сети (веса связей) настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки этих параметров.

Свойство нейронной сети обучаться на примерах позволяет упростить задание условий для решения конкретной задачи, по сравнению с системами, которые работают по заранее готовому алгоритму, составленному программистами. Обучение нейросетей подразделяют на 2 группы: обучение с учителем и без учителя.

1.Обучение нейронной сети с учителем

В этом типе обучения для каждого набора входных данных из обучающего множества существует требуемое значение выходных данных. Веса связей сети изменяют до тех пор, пока для каждого входного вектора не будет получен приемлемый уровень отклонения выходного вектора от целевого.

2. Обучение нейронной сети без учителя

В таком случае обучающее множество состоит лишь из входных векторов. Алгоритм обучения нейронной сети корректирует веса связей так, чтобы выходные вектора нейросетей для достаточно близких входных векторов были достаточно близки.

Один из самых распространенных алгоритмов обучения с учителем - **алгоритм обратного распространения ошибки** или **back propagation**.

Для начала рассмотрим такое понятие, как ошибка нейронной сети. Определим ее как некую функцию $E(X, Y, w)$, где X - входной вектор значений, Y- ожидаемый результат, w- веса связей нейросети.

Тогда задачу обучения нейросети можно описать как нахождение таких весов w^*, что $w^* = argmin_w E(X, Y, w)$.

Т.е. задача состоит в том, чтобы найти такие веса w, при которых функция E минимальна.

Способов нахождения таких весов много, например, можно воспользоваться просто случайным перебором, генетическим алгоритмом оптимизации, но самым оптимальным для задачи обучения нейросети будет алгоритм градиентного спуска (или его модификации).

Суть в данного алгоритма заключается в следующем: на каждой итерации вычисляем антиградиент функции E по весам w, сдвигаем вектор весов на $\alpha\nabla_w$, где $\alpha \in R$. В таком случае на какой-то итерации придем к минимуму функции $E(w)$, что и требуется сделать.

При обучении нейронной сети входные данные многократно подаются нейросети, и на каждом таком шаге веса корректируются.

Часто после завершения обучения для оценки точности на вход подается контрольная выборка данных - т.е. та, которая не встречалась в обучающей выборке, на данном этапе не производится коррекция весов, а только лишь вычисляется ошибка.

1.5 Нейросети в задачах анализа текста

Для задачи анализа текста применяется большое количество типов нейронных сетей, например, сверточные нейронные сети (CNN) или рекуррентные нейронные сети (RNN).

Стоит отметить, что, как правило, перед применением к исходным данным алгоритмов обучения нейросети, эти данные обрабатывают и подвергают изменениям, которые позволяют улучшить результаты обучения.

Несмотря на то, что задачи анализа текста можно решать многими способами, этап предобработки данных у всех решений примерно один и тот же, а именно:

1. Удаление пунктуации

   Пунктуация позволяет в более полном объеме понять смысл фразы или предложения. Однако она не имеет почти никакого смысла для нейронной сети, так как нейросеть анализирует данные по количеству отдельно взятых слов их весу, а не смыслу предложения. Поэтому для исходных данных удаляют всю пунктуацию, например, фраза “What time is it?” будет изменена на “What time is it”.

2. Удаление стоп-слов (stopwords)
   Стоп-словами (stopwords) принято обозначать слова, которые формируют структуру предложения, но не несут какой-то смысловой ценности для принятия решения нейросетью (например, стоп-словами являются междометия), поэтому такие слова форматируются из начальных данных. Например, фраза “Yeah it is a good place to visit” будет трансформирована в набор слов “good, place, visit”.
3. Удаление суффиксов (Stemming)
   В абсолютном большинстве случаев под разными склонениями и падежами слов подразумевается одинаковый смысл, в таком случае лучше, чтобы нейросеть воспринимала такие слова как одно. Для этого из однокоренных слов удаляют суффиксы.
   Например, “enable, enabled” будет обработано как “enabl”
4. Лемматизация (Lemming)
   Более точная, но более медленная версия удаления суффиксов. Отличие состоит в том, что лемматизация выделяет корень слова, этот процесс работает на основе анализа словаря, что повышает общую точность подхода, однако это более трудоемкая операция.
   При таком подходе “enable, enabled” будет обработано как “enable”
5. Токенизация и векторизация
   Токенизация - процесс разделения фразы на отдельные компоненты с использованием вышеперечисленных принципов.
   Векторизация - способ представления слов в виде численного типа, чтобы математические алгоритмы имели возможность работать с ними.

Помимо вышеперечисленных практик, хорошим способом увеличить точность является добавление в исходные данные новых признаков - например, длину фразы, уникальность слов. Также, несмотря на то, что знаки пунктуации удаляются из состава слов и фраз, введение такого признака, как “среднее количество знаков пунктуации” часто повышает точность результатов обучения нейросети.

2. Задача классификации спам-сообщений

2.1 Описание задачи

В повседневной жизни нам приходится сталкиваться с большим объемом поступающей информации, в числе которого могут оказаться смс-сообщения, которые приходят на личный телефон. Но далеко не каждое сообщение несет в себе посыл передать какую-то информацию лично от человека к человеку. Часто в виде сообщений присылают рекламу, которую большинство людей готовы пропустить в процессе просмотра сообщений. Поэтому появляется задача классификации, или отделения так называемого спама от обычных сообщений.

Решение данной задачи с помощью нейронных сетей будет рассмотрено далее.

2.2 Решение задачи

В качестве начального набора данных был выбран набор смс-сообщений на английском языке [5], которые уже изначально классифицированы как спам или не спам. Данные представлены в виде таблицы и структурированы следующим образом:

Таблица 1. Пример структуры выбранного набора данных

spam	Free entry in 2 a wkly comp to win FA Cup...
not spam	Go until jurong point, crazy.. Available only ...
not spam	Nah I don't think he goes to usf, he lives aro...
...	...

Всего сообщений в наборе данных - 5572. Спам-сообщения составляют примерно 15% от всех сообщений.

Перед обучением нейронной сети, данные следует подготовить: прежде всего, случайным образом удалим часть сообщений из группы “not spam” для того, чтобы сбалансировать набор данных.

Далее отделим часть данных, на которых нейросеть обучаться не будет, на них после обучения можно будет проверить точность работы.
Далее избавимся от пунктуации и стоп-слов, после чего проведем процесс токенизации и векторизации над данными.
После этого обучим нейросеть.

2.3 Результат. Выводы

В результате обучения нейросети, получим точность ~94% на оставленной ранее тестовой выборке.
Можем проверить результат обучения нейросети собственноручно, подав на вход вымышленное смс-сообщение. Для проверки возьмем такие сообщения:

- “Hi, Harry. I'm busy at the moment. Studying calculus. Call me later”
- “I don't really think I can attend the show”
- “Almost free Dominos pizza on Tuesdays. 2 for the price of 1. Call 777888”
- “Want to have a lot of cash almost for free? Follow the link”

Первые два сообщения не являются спамом, а вторые два - наоборот, являются.
Получим следующие результаты:

```
Spam Probability: 2.0709961652755737%
Spam Probability: 0.288623571395874%
Spam Probability: 92.25412607192993%
Spam Probability: 80.2541732788086%
```

Рисунок 4. Вывод программы

Полученные результаты подтверждают вывод, что нейросеть обучилась правильно. Разницу в результатах для спам-сообщений можно объяснить тем, что в первом сообщении содержится большее количество слов, свойственных для рекламных сообщений, нежели во втором, что повышает их “вес” в процессе принятия решения нейросети. Но даже не смотря на это, результат для второго спам-сообщения достаточно убедительный для задачи классификации.
В качестве результата работы приведем графики с информацией о процессе обучения нейронной сети:

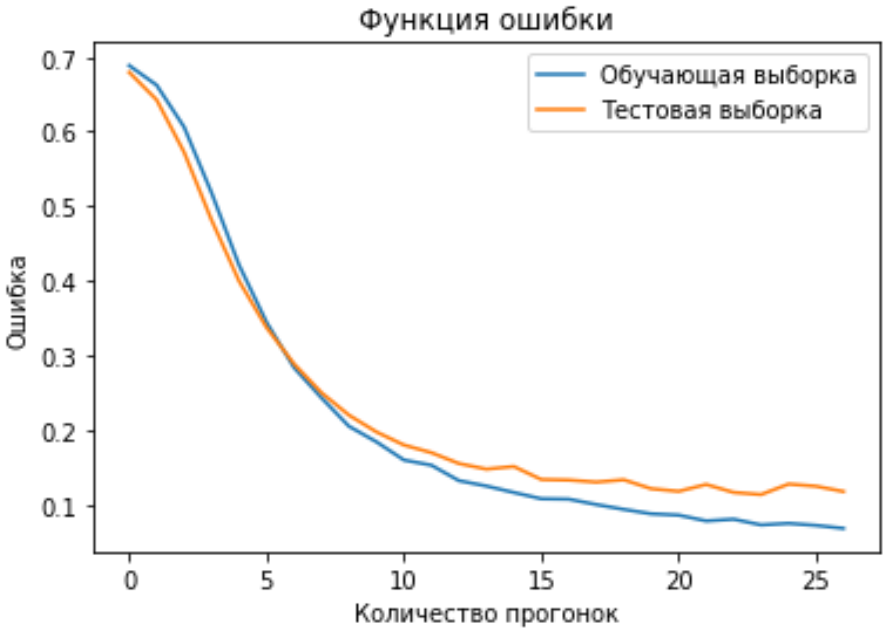

Рисунок 5. Значение функции ошибки

На графике видно, что за первые 15 прогонок одновременно уменьшается и ошибка на обучающей выборке, и ошибка на тестовой выборке. Однако после этой отметки, ошибка на обучающей выборке продолжает уменьшаться, а ошибка на тестовой выборке начинает колебаться примерно в тех же значениях. Это можно объяснить тем, что нейронная сеть продолжает подстраивать веса для лучшего результата на обучающей выборке, в то время, как общее качество обучения нейросети уже достигнуто, поэтому ошибка начинает колебаться для тестовой выборки.

Стоит отметить, что если мы продолжим обучать нейросеть и далее, то как результат получим эффект так называемого “переобучения” - нейросеть слишком хорошо подстроится под обучающую выборку, из-за чего ошибки на ней почти не будет, в то время как для тестовой выборки ошибка может достигать очень больших размеров. Чтобы избавиться от проблемы переобучения нейронной сети, используют множество техник, начиная от ограничения количества прогонок входных данных, заканчивая разделением данных на несколько частей, и последующем обучением на этих частях и проверкой на незадействованных в обучении частей (кросс-валидация).

Данная задача была решена с помощью обычной многослойной нейронной сети. В то же время известно, что такие разновидности нейронных сетей, как конволюционные нейронные сети и рекуррентные нейронные сети зарекомендовали себя лучше в задачах классификации в силу более сложной, но и более функциональной архитектуры. Поэтому можно утверждать, что с большой долей вероятности, если применить к исходным данным правильно настроенную конволюционную и рекуррентную нейронную сеть, то результат получится лучше, чем в данной работе, при использовании обычной многослойной нейронной сети.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены основные принципы работы нейронной сети, а также их приложения для задачи анализа текста. Была рассмотрена задача распознавания спама в смс-сообщениях, а также построена программная реализация решения этой задачи с помощью языка программирования Python 3 [6], а также фреймворков для машинного обучения: tensorflow, sklearn. Был проведен анализ результатов практической работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Make Your Own Neural Network: A Gentle Journey Through the Mathematics of Neural Networks / T. Rashid - CreateSpace Independent Publishing Platform, 2016 - 222 p.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition / Aurélien Géron - O'Reilly Media, Inc., 2019 - 856 p.
3. Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning 1st Edition / Benjamin Bengfort, Rebecca Bilbro, Tony Ojeda - O'Reilly Media, Inc., 2018 - 332 p.
4. Фреймворк для задач машинного обучения с помощью нейросетей Tensorflow / Google, 2020 - Документация: www.tensorflow.org/guide
5. Открытый датасет смс-сообщений для задачи классификации спама / University of California Irvine, 2012 - ссылка: www.archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
6. GitHub репозиторий с программной реализацией решения задачи классификации спама в смс-сообщениях - ссылка: https://github.com/desobolevsky/UniCourseworks/tree/main/Coursework_Fall2020