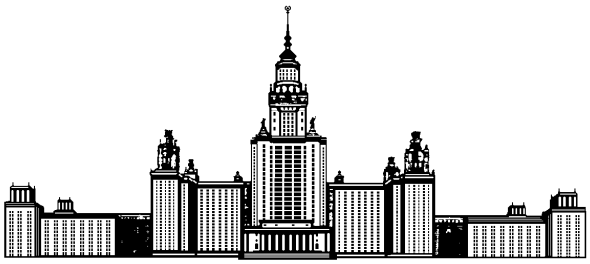

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра Математических Методов Прогнозирования

Рысьмятова Анастасия Александровна

# Использование сверточных нейронных сетей для задачи классификации текстов

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Научный руководитель:**
д.ф.-м.н., профессор
А.Г. Дьяконов

Москва, 2016

# Содержание

**Аннотация**

Сверточные нейронные сети — мощный инструмент машинного обучения, который нацелен на эффективное распознавание и классификацию изображений. Успех применения сверточных нейронных сетей для изображений породил множество попыток использования этого инструмента в других задачах.

В данной работе исследованы основные методы использования сверточных нейронных сетей для задачи классификации текстов. Выполнены эксперименты на текстовых данных большого объема, показавшие, что сверточные нейронные сети для задачи классификации текстов позволяют достичь качества, аналогичного или лучшего в сравнении с традиционными методами.

# 1 Введение

Задача классификации текстов становится все более актуальной в связи с постоянно растущим объемом информации в интернете и потребностью в ней ориентироваться. К примеру, классификация текстов необходима для решения следующих задач:

1. Борьба со спамом.

   Спам — это нежелательные рассылки, которые могут приходить на адрес электронной почты. Они могут содержать рекламные предложения или компьютерные вирусы. Задача борьбы со спамом заключается в том, чтобы классифицировать все письма на два класса: спам и не спам.

2. Распознавание эмоциональной окраски текстов.

   Задача заключается в том, чтобы оценить мнение автора по отношению к объектам, например на основе отзывов об этих объектах. Часто такую задачу необходимо решать для выдачи релевантных рекомендаций.

3. Разделение сайтов по тематическим каталогам.

   Данная задача решается поисковыми системами и предусматривает обработку документов и отнесение их к одной из нескольких категорий, перечень которых заранее задан.

4. Персонификация рекламы.

   Контекстная реклама является основным источником дохода IT компаний. Она отображается посетителям интернет-страницы, сфера интересов которых потенциально совпадает или пересекается с тематикой рекламируемого товара либо услуги, целевой аудитории, что повышает вероятность их отклика на рекламу. Сфера интересов определяется по тексту интернет-страниц просмотренных пользователем.

В связи с важностью данной задачи, по ее решению проводятся множество соревнований по машинному обучению с ценными призами, исследуются новые методы для достижения лучшего качества классификации.

В данной работе рассматриваются основные методы классификации текста, а так же недавно предложенный в статье [13] посимвольный подход с использованием сверточных нейронных сетей. В работе реализована сверточная нейронная сеть с посимвольным подходом[13] осуществляющая классификацию текстов и показано, что на данных большого объема посимвольный подход показывает лучшее качество классификации в сравнении с традиционными методами.

Основным вкладом данной работы является исследование влияния предобработки текста на качество классификации с помощью сверточных нейронных сетей с посимвольным подходом.

# 2 Постановка задачи классификации

Классификация[14] решает следующую задачу. Задано конечное множество классов и имеется множество объектов, для конечного подмножества которых известно к какому классу они относятся. Это подмножество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Классифицировать объект — значит, указать номер (или наименование класса), к которому относится данный объект.

В задачи классификации текстов объекты — это текстовые документы.

Запишем формальную постановку задачи классификации текстов.

$D = \{d_1, ..., d_n\}$ — множество текстовых документов. Каждый документ $d \in D$ представляет собой последовательность слов $W_d = (w_1, \ldots, w_{n_d})$, $n_d$ — длина документа $d$.

$Y = \{y_1, ..., y_n\}$ — конечное множество меток классов.

$y^* : D \to Y$ — неизвестная целевая зависимость, значения которой известны только на объектах конечной обучающей выборки $D^m = \{(d_1, y_1), \ldots, (d_m, y_m)\}$.

Требуется построить алгоритм $a : D \to Y$, способный классифицировать произвольный объект $d \in D$.

# 3 Традиционные методы машинного обучения для классификации текстов

Обычно задачу классификации текстов решают с помощью следующих этапов :

1. Предобработка текстов.
2. Перевод текстов в вещественное пространство признаков, где каждому документу будет сопоставлен вектор фиксированной длины.
3. Выбор алгоритма машинного обучения для классификации.

Опишем подробнее каждый из этапов.

## 3.1 Предобработка текстов

Все тексты на естественном языке имеют большое количество слов, которые не несут информации о данном тексте. К примеру, в английском языке такими словами являются артикли, в русском к ним можно отнести предлоги, союзы, частицы. Данные слова называют шумовыми или стоп-словами. Для достижения лучшего качества классификации на первом этапе предобработки текстов обычно необходимо удалять такие слова.

Второй этап предобработки текстов — приведение каждого слова к основе, одинаковой для всех его грамматических форм. Это необходимо, так как слова несущие один и тот же смысл могут быть записаны в разной форме. Например, одно и то же слово может встретиться в разных склонениях, иметь различные приставки и окончания.

## 3.2 Перевод текста в векторное представление

Большинство современных алгоритмов машинного обучения ориентированы на признаковое описание объектов, поэтому все документы обычно переводят в вещественное пространство признаков. Для этого используют идею о том, что за принадлежность документа к некоторому классу отвечают слова, а тексты из одного класса будут использовать много схожих слов.

Наиболее известные способы, позволяющие осуществить перевод текста в пространство признаков, основаны на статистической информации о словах. При их использовании каждый объект переводится в вектор, длина которого равна количеству используемых слов во всех текстах выборки.

### 3.2.1 Bag of Words

Bag of Words [3](мешок слов) — модель перевода текста в векторное представление. Основное предположение данного метода — порядок слов в документе не важен, а коллекцию документов можно рассматривать как простую выборку пар «документ–слово» $(d, w)$, где $d \in D,\ w \in W_d$. В Bag of Words все документы представляются в виде матрицы $T = (t)_{d,w}$, каждая строка в которой соответствует отдельному документу или тексту, а каждый столбец — определенному слову. Элемент $t_{d,w}$ соответствует количество вхождений слова $w$ в документ $d$.

### 3.2.2 Bag of Words & TF IDF

Это наиболее популярный способ перевода текста в векторное представление. Как и в методе Bag of Words все документы представляются в виде матрицы $T = (t)_{d,w}$, но элемент $t_{d,w}$ соответствует функции TF-IDF$(w, d, D)$ слова $w \in W_d$ в документе $d \in D$.

**Определение 3.1.** *TF-IDF [10] — это статистическая мера, используемая для оценки важности слова в контексте документа. Вычисляется по формуле:*

$$\text{TF-IDF}(w, d, D) = \text{TF}(w, d) \times \text{IDF}(w, D)$$

TF — частота слова, оценивает важность слова $w_i$ в пределах отдельного документа.

$$\text{TF}(w, d) = \frac{n_i}{\sum_k n_k}$$

$n_i$ — число вхождений слова $i$ в документ.

$\sum_k n_k$ — общее число слов в данном документе.

IDF — обратная частота документа. Учёт IDF уменьшает вес широко употребляемых слов.

$$\text{IDF}(w, D) = \log \frac{|D|}{|(d_i \supset w_i)|}, \text{ где}$$

$| D |$ — количество документов в корпусе.

$| (d_i \supset w_i) |$ — количество документов, в которых встречается слово $w_i$.

### 3.2.3 Bag of Ngrams & TF IDF

Часто информацию в тексте несут не только отдельные слова, но и некоторая последовательность слов. Например, фразеологизмы — устойчивые сочетание слов значение которых не определяется значением входящих в них слов, взятых по отдельности. Например, речевой оборот «Как рыба в воде» означает чувствовать себя уверенно, очень хорошо в чем-либо разбираться. Смысл данного выражения будет передан неверно, если учитывать его слова по отдельности.

Для того чтобы учесть такие особенности языка предлагается при переводе текстов в векторное представление учитывать помимо слов, N-граммы.

N-граммы[1] — это последовательности из N слов. К примеру, для текста «мама мыла раму» получаем биграммы «мама мыла» и «мыла раму». В задаче классификации текстов N-граммы являются индикаторами того, что данные N слов встретились рядом

Метод *Bag of Ngrams & TF IDF* аналогичен методу *Bag of Words & TF IDF*, только вектор признаков для каждого документа помимо TF IDF слов содержит TF IDF всех последовательностей из N слов.

## 3.3 Выбор алгоритма классификации

Полученное признаковое пространство в данном методе будет сильно разряженным и будет иметь высокую размерность за счет того, что различных слов встречающихся во всей выборке обычно много. Из-за этого для данной задачи чаще всего используют линейные методы машинного обучения.

## 3.4 Проблемы традиционного метода классификации текстов

1. Необходимо выбрать метод перевода текста в векторное представление, ведь обычно для различных задач наилучшее качество классификации показывают различные методы.

2. Полученное признаковое пространство будет иметь высокую размерность, а так же будет сильно разряженным.

3. Чаще всего для улучшения качества классификации необходимо удалить из текста стоп-слова. При удалении различного набора стоп-слов получается разный результат работы алгоритма.

4. Для применения данного метода необходимо применения стемминга или лемматизации, так как слова имеющие разное склонение несут один и тот же смысл.

# 4 Нейронные сети

Попытки воспроизвести способность биологических нервных систем обучаться и исправлять ошибки привели к созданию искусственных нейронных сетей. Искусственные нейронные сети представляют собой семейство моделей, построенных по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

Понятие искусственной нейронной сети было предложено ещё в 1943 году У. Маккалоком и У. Питтсом в статье [8]. В частности, ими была предложена модель искусственного нейрона.

Чтобы отразить суть биологических нейронных систем, искусственный нейрон строится следующим образом. Он получает входные сигналы (исходные данные либо выходные сигналы других нейронов нейронной сети) через несколько входных каналов. Каждый входной сигнал проходит через соединение, имеющее определенный вес. С каждым нейроном связано определенное пороговое значение. Вычисляется взвешенная сумма входов, из нее вычитается пороговое значение и в результате получается величина активации нейрона . Сигнал активации преобразуется с помощью функции активации и в результате получается выходной сигнал нейрона.

На Рис.1 приведен пример искусственного нейрона.

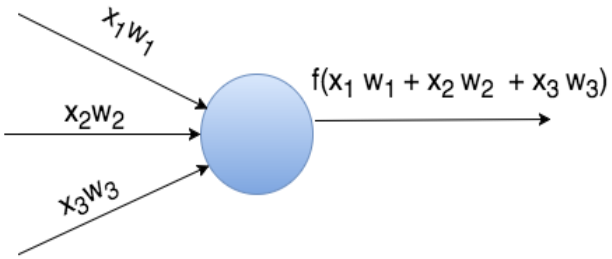


Рис. 1: Искусственный нейрон

$x_i$— входной сигнал

$w_i$— вес входного сигнала

$f(\cdot)$— функция активации

## 4.1 Функции активации

В данном разделе описаны используемые в работе функций активаций для нейронных сетей.

- Сигмоидная: $f(s) = \dfrac{1}{1+e^{-s}}$
- Линейная: $f(s) = s$
- Положительно линейная (Relu): $f(s) = \max(0, s)$
- Софтмакс (Softmax): $f(s)_j = \dfrac{e^{s_j}}{\sum_{k=1}^{K} e^{s_k}}$ для $j = 1, ..., K$

## 4.2 Функция потерь

Введем обозначения: $X$— множество описаний объектов, $Y$— множество допустимых ответов. Предполагается, что существует неизвестная целевая зависимость — отображение $y^* : X \to Y$, значения которой известны только на объектах конечной обучающей выборки $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.

Вводится функция потерь $\mathcal{L}(y, y')$, характеризующая величину отклонения ответа $y$ от правильного ответа $y' = y^*(x)$ на произвольном объекте $x \in X$. Тогда эмпирический риск [14] — функционал качества, характеризующий среднюю ошибку на обучающей выборке:

$$Q(a, X^m) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y_i, y^*(x_i))$$

В процессе обучения нейронная сеть настраивает веса $W$, минимизируя эмпирический риск.

При решении задачи многоклассовой классификации на выходе нейронной сети необходимо получить вероятность принадлежности объекта каждому из классов. В этом случае в качестве функции потерь обычно используется кросс-энтропия

$$\mathcal{L}(y_i, y^*(x_i)) = -\sum_{j=1}^{K} y_{ij}^* \log y_{ij}$$

$K$ — количество меток классов в задаче.

В данной работе для классификации текстов с помощью нейронных сетей используется описанная функция потерь.

# 5 Сверточные нейронные сети

С появлением больших объемов данных и больших вычислительных возможностей стали активно использоваться нейронные сети. Особую популярность получили сверточные нейронные сети, архитектура которых была предложена Яном Лекуном [12] и нацелена на эффективное распознавание изображений. Свое название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения. В архитектуру сети заложены априорные знания из предметной области компьютерного зрения: пиксель изображения сильнее связан с соседним (локальная корреляция) и объект на изображении может встретиться в любой части изображения.

Особое внимание светрочные нейронные сети получили после конкуса ImageNet, который состоялся в октябре 2012 года и был посвящен классификации объектов на фотографиях. В конкурсе требовалось распознавание образов в 1000 категорий. Победитель данного конкурса — Алекс Крижевский, используя сверточную нейронную сеть, значительно превзашел остальных участников [6].

Успех применения светрочных нейронных сетей к классификации изображений привел к множеству попыток использовать данный метод к другим задачам. В последнее время их стали активно использоваться для задачи классификации текстов.

## 5.1 Архитектура сверточной нейронной сети

Сверточная нейронная сеть обычно представляет собой чередование сверточных слоев (convolution layers), субдискретизирующих слоев (subsampling layers) и при наличии полносвязных слоев (fully-connected layer) на выходе. Все три вида слоев могут чередоваться в произвольном порядке [12].

В сверточном слое нейроны, которые используют одни и те же веса, объединяются в карты признаков (feature maps), а каждый нейрон карты признаков связан с частью нейронов предыдущего слоя. При вычислении сети получается, что каждый нейрон

выполняет свертку некоторой области предыдущего слоя (определяемой множеством нейронов, связанных с данным нейроном).

Пример архитектуры сверточной нейронной сети представлен на Рис. 2.

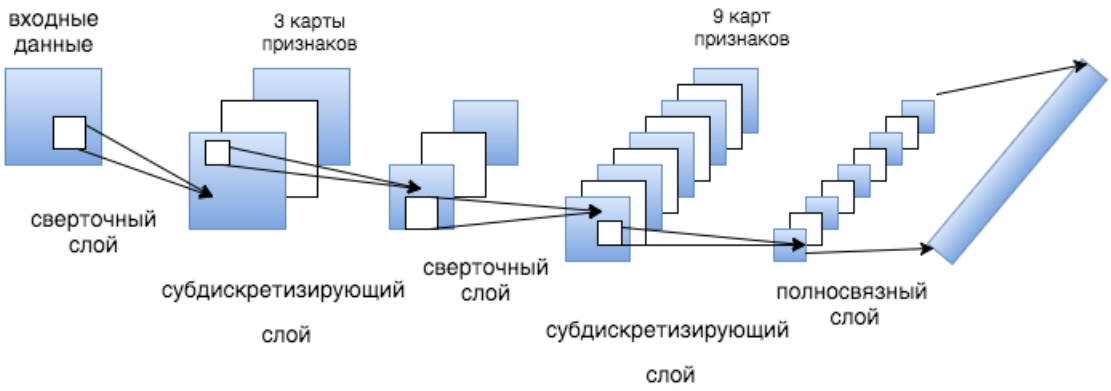


Рис. 2: Архитектура сверточной нейронной сети

### 5.1.1 Полносвязный слой

Слой в котором каждый нейрон соединен со всеми нейронами на предыдущем уровне, причем каждая связь имеет свой весовой коэффициент. На Рис.3 показан пример полносвязного слоя.

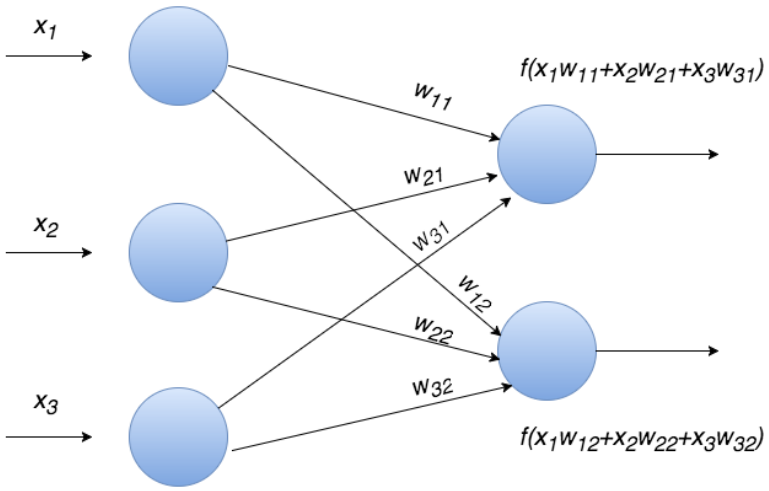


Рис. 3: Полносвязный слой

$w_{ij}$ — весовые коэффициенты.

$f(\cdot)$ — функция активации.

### 5.1.2 Сверточный слой

В отличие от полносвязного, в сверточном слое нейрон соединен лишь с ограниченным количеством нейронов предыдущего уровня, т. е. сверточный слой аналогичен применению операции свертки, где используется лишь матрица весов небольшого размера (ядро свертки), которую «двигают» по всему обрабатываемому слою.

Еще одна особенность сверточного слоя в том, что он немного уменьшает изображение за счет краевых эффектов.

На Рис. 4 показан пример сверточного слоя с ядром свертки размера $3 \times 3$.

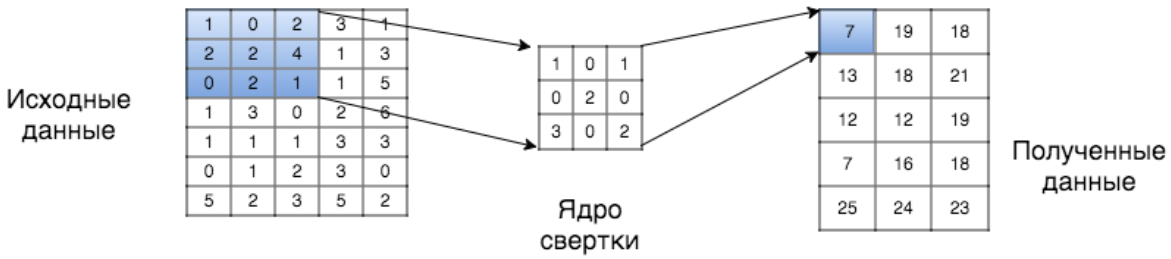


Рис. 4: Сверточный слой

### 5.1.3 Субдискретизирующий слой

Слои этого типа выполняют уменьшение размерности (обычно в несколько раз). Это можно делать разными способами, но зачастую используется метод выбора максимального элемента (max-pooling) — вся карта признаков разделяется на ячейки , из которых выбираются максимальные по значению.

На Рис. 5 показан пример субдискретизирующего слоя с методом выбора максимального элемента.

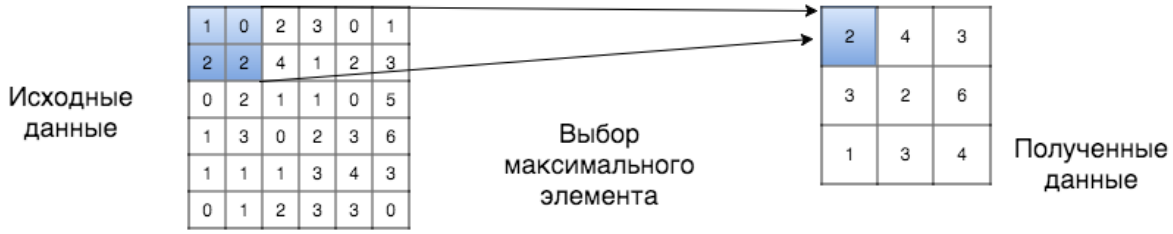


Рис. 5: Субдискретизирующий слой

### 5.1.4 Dropout слой

Dropout слой (dropout регуляризация) [14] способ борьбы с переобучением в нейронных сетях, обучение которых обычно производят стохастическим градиентным спуском, случайно выбирая некоторые объекты из выборки. Dropout регуляризация заключается в изменение структуры сети: каждый нейрон выбрасывается с некоторой вероятностью $p$. По такой прореженной сети производится обучение, для оставшихся весов делается градиентный шаг, после чего все выброшенные нейроны возвращаются в нейросеть. Таким образом, на каждом шаге стохастического градиента мы настраиваем одну из возможных $2^N$ архитектур сети, где под архитектурой мы понимаем структуру связей между нейронами, а через N обозначаем суммарное число нейронов. При тестировании нейросети нейроны уже не выбрасываются, но выход каждого нейрона умножается на $(1 - p)$ — благодаря этому на выходе нейрона мы будем получать матожидание его ответа по всем $2^N$ архитектурам. Таким образом, обученную с помощью dropout-регуляризации нейросеть можно рассматривать как результат усреднения $2^N$ сетей.

## 5.2 Модели использования сверточной нейронной сети для классификации текстов

В данном разделе будут описаны основные подходы использования сверточных нейронных сетей для задачи классификации текстов.

### 5.2.1 Посимвольный подход

Посимвольных подход для классификации текстов с помощью сверточных нейронных сетей был предложен в статье [13]. Опишем данный метод подробнее.

Назовем алфавитом упорядоченный набор символов. Пусть выбранный алфавит состоит из $m$ символов. Каждый символ алфавита в тексте закодирован с помощью $1-m-$ кодировки. (т. е. каждому символу будет сопоставлен вектор длины $m$ элемент которого равен единице, в позиции равной порядковому номеру символа в алфавите, а нулю во всех остальных позициях.) Если в тексте встретиться символ, который не вошел в алфавит, то необходимо закодировать его вектором длины $m$ состоящим из одних нулей. Из текста выбираются первые $\ell$ символов. Параметр $\ell$ должен быть большим, чтобы в первых $\ell$ символах содержалось достаточно информации для определения класса всего текста.

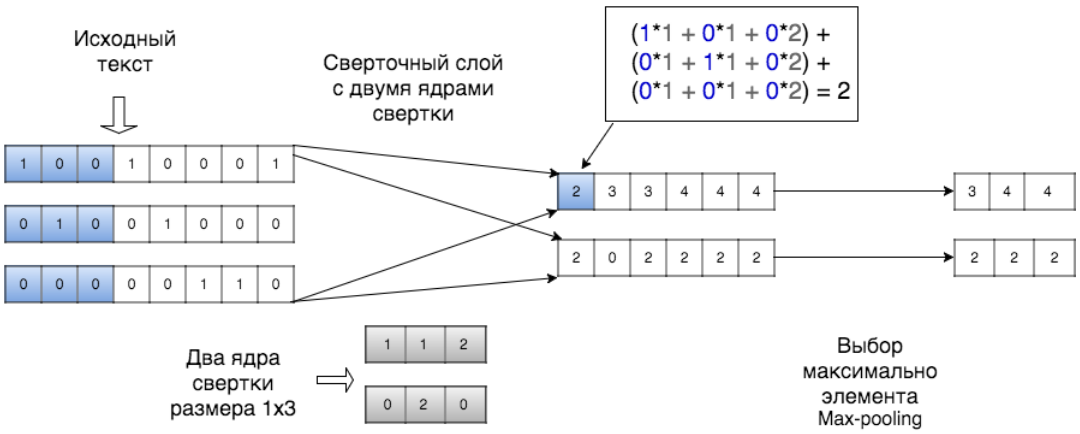


Рис. 6: Посимвольный подход

Далее полученные векторы составляются в матрицу размера $m \times \ell$, в которой в каждый столбец будет иметь не более одной единицы. Каждая строка полученной матрицы используется как отдельная карта признаков. На вход светрочной нейронной сети подается $m$ карт признаков размера $1 \times \ell$ аналогично изображению. Архитектуру сети необходимо выбирать исходя из задачи. На Рис. 6 приведен пример посимвольного подхода для $\ell = 6$, $m = 3$. В примере показан один сверточный и один субдискретизирующий слой.

Опишем формально данный подход.

Пусть $x_i$ — вектор $i$-го символа в тексте.

$$x_{1:\ell} = x_1 \oplus x_2 \oplus ... \oplus x_\ell$$

Здесь $\oplus$ операция объединения векторов.

**Сверточный слой:**

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$
$$c = [c_1, c_2, ..., c_{n-h+1}]$$

$f$ — функция активации нейронной сети

$b$ — константа

**MAX-pooling слой:**

$$\widehat{c} = max\{c\}$$

**Dropout слой:**

$$y = w(z \circ r) + b$$

Здесь $\circ$ — посимвольное умножение.

$r$ — вектор состоящий из нулей и едениц.

В статье [13] были приведены эксперименты, которые показали, что описанный подход с высокой точностью классифицирует тексты, по сравнению с большинством других известных на данный момент методов классификации текстов, если размеры выборки достаточно велики. Так на выборке размером 1400000 объектов сверточная нейронная сеть с посимвольным подходом дала качество классификации по метрике accuracy — 0.712, а методом *Bag of words* удалось достичь лишь — 0.689.

### 5.2.2 Подход с использованием кодирования слов

Подход был описан в статье [5]. В данном подходе каждому слову в тексте сопоставляется вектор фиксированной длины, затем из полученных векторов для каж-

дого объекта выборки составляется матрица, которая аналогично изображениям подается на вход сверточной нейронной сети. На Рис. 7 приведен пример сверточной нейронной сети с использованием кодирования слов.

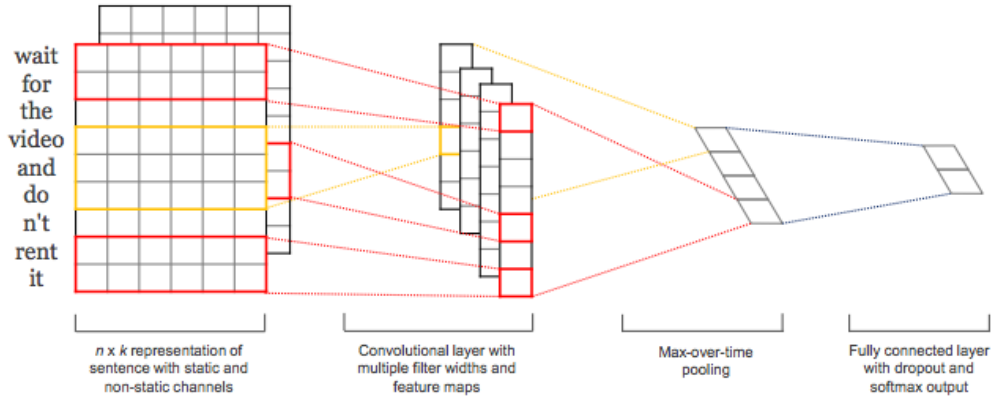


Рис. 7: Посимвольный подход [5]

Для экспериментов в статье [5] была реализована нейронная сеть с одним сверточным, одним субдискритезирующим и одним полносвязным слоем. Данная нейронная сеть использовалась для классификации текстов небольшого размера, состоящих из одного предложения.

## 5.3 Методы перевода слова в вектор фиксированной длины

В данном разделе будут описаны наиболее известные методы представления слова с помощью вектора фиксированной длины.

### 5.3.1 One-hot кодировка

В данном методе каждое слово кодируется с помощью вектора фиксированной длины, равной количеству используемых слов в выборке. Каждый вектор состоит из нулей и одной единицы.

### 5.3.2 Word2Vec

Word2Vec[2] — технология от компании Google, которая заточена на статистическую обработку больших массивов текстовой информации. Он собирает статистику о появлении слов в данных, удаляет наиболее редко встречаемые и часто встречаемые слова после чего методами нейронных сетей решает задачу снижения размерности и выдает на выходе компактные векторные представления слов заранее определенной длины. При этом Word2Vec максимизирует косинусную меру близости между векторами слов, которые встречаются в близких контекстах и минимизирует косинусную меру между словами которые не встречаются рядом.

**Определение 5.1.** *Косинусная мера близости (косинусное сходство) – это мера сходства между двумя векторами. Косинусное сходство между векторами $A$ и $B$ вычисляется по формуле:*

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

В Word2Vec можно использовать две различных архитектуры нейронной сети для перевода слова в вектор: Continuous Bag of Words и Skipgram.

**Continuous Bag of Words (CBOW)** — мешок слов. В данном подходе весь текст просматривается окном ширины $2h + 1$ и в каждом окне нейронная сеть предсказывает центральное слово окна $w_t$, по всем остальным словам в этом окне $w_i$, где $i \in [t - h, 0) \cup (0, t + h]$.

Рассмотрим подробнее как устроена нейронная сеть для задачи снижения размерности в CBOW. Для начала рассмотрим случай когда хотим предсказать слово только по одному слову из текста. Обозначим $V-$ количество различных слов в данных (объем словаря), $N-$ длина получаемого вектора, соответствующего предсказываемому слову. Нейронная сеть состоит из трех слоев. На входном уровне (input) и выходном уровне (output) — V нейронов, на скрытом уровне (hidden) — N нейронов. Оба слоя нейронной сети полносвязные. На Рис. 8 приведен пример архитектуры сети.

Пусть вектор для входного слова $x = (x^1, x^2, ..., x^V)$. Если входное слово соответствует номеру $k$ в словаре, то $x^k = 1$ и $x^{k'} = 0$ для всех $k \neq k'$. Функция активации

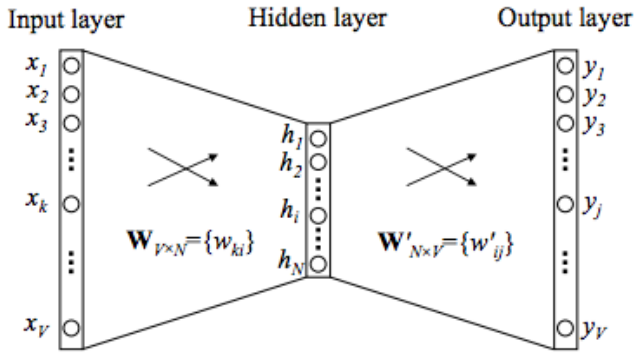


Рис. 8: Архитектура сети CBOW [11]

на скрытом уровне — линейная, на выходном уровне в зависимости от выбранных параметров — софтмакс (Softmax)

Веса между входным и скрытым уровнем можно представить в виде матрицы $W_{V\times N} = w_{ki}$, тогда вектор выходов нейронов на срытом уровне можно представить в виде произведения вектора на матрицу

$$h = x^T W$$

Веса между скрытым и выходным уровнем можно представить в виде матрицы $W'_{N\times V} = w'_{ij}$. Обозначим $w'_j-$ столбец $j$ матрицы $W'$. Тогда входной сигнал для $j-$го нейрона на выходном уровне:

$$u_j = w'^T_j h$$

Если используется на выходном уровне используется функция активации softmax, то выходной сигнал на $j-$ ом нейроне выходного слоя

$$y_j = \frac{exp(u_j)}{\sum_{j'=1}^{V} exp(u_{j'})} = p(w_O|w_I)$$

$w_I-$вектор соответствующий входному слову.

$w_O-$вектор соответствующий выходному слову.

Обучаясь, нейронная сеть максимизирует $y_{j^*}$, где $j^*-$ номер предсказываемого слова в словаре, или что тоже самое в контексте данной задачи максимизирует вероятность выходного слова $w_O$ при условии, что известно входное слово $w_I$.

$$\max y_{j^*} = \max \log y_{j^*} = u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) = \max p(w_O|w_I)$$

После обучения нейронной сети столбцы матрицы $W'$ — вектора длины $N$ будут результатом работы алгоритма CBOW технологии Word2Vec.

В общем случае, когда хотим предсказать слово по $C$ словам из текста, на входном уровне будет $C \times V$ нейронов. Выходной сигнал на скрытом уровне вычисляется по формуле:

$$h = \frac{1}{C} W(x_1 + x_2 + ... + x_C)$$

Здесь $x_i = (x_i^1, x_i^2, ..., x_i^V)$−вектор соответствующий $i$−ому входному слову. На Рис. 9 приведен пример архитектуры сети для общего случая.

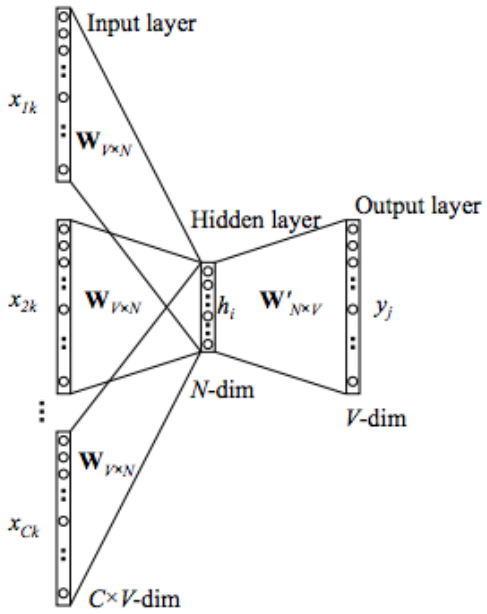


Рис. 9: Архитектура сети для общего случая CBOW [11]

Обучаясь, нейронная сеть максимизирует $y_{j^*}$, где $j^*$− номер предсказываемого слова в словаре, или, что тоже самое в контексте данной задачи, максимизирует вероятность выходного слова $w_O$ при условии, что известны входное слово $w_{I,1}, w_{I,2}, ..., w_{I,C}$.

$$\max y_{j^*} = \max \log y_{j^*} = u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) = \max p(w_O | w_{I,1}, w_{I,2}, ..., w_{I,C})$$

Остальные формулы вычисляются аналогично случаю когда хотим предсказать слово только по одному слову из текста.

**Skipgram** — В данном подходе весь текст просматривается окном ширины $2h+1$ и в каждом окне нейронная сеть предсказывает слова в этом окне $w_i$, где $i \in [t-h, 0) \cup$

$(0, t+h]$ по центральному слову окна $w_t$. На Рис. 10 приведен пример архитектуры сети для skipgram модели.

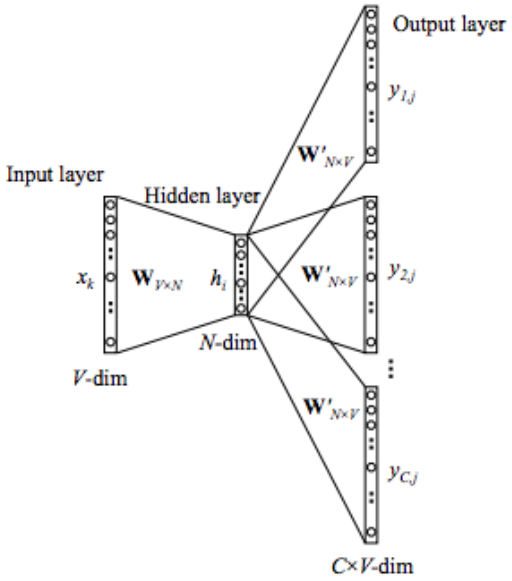


Рис. 10: Архитектуры сети для skipgram модели [11]

В данной модели выходной сигнал на $j-$ом выходном нейроне для предсказываемого слова под номером $c$:

$$y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{j'})} = p(w_{c,j} = w_{O,c}|w_I)$$

Предположим, что нейронная сеть предсказывает C слов. Обучаясь, она максимизирует $y_{j^*}$, где $j^*-$ вектор длины C, номера предсказываемых слов в словаре, или, что тоже самое в контексте данной задачи, максимизирует вероятность выходных слов $w_{O,1}, w_{O,2}, ..., w_{O,C}$ при условии, что известно входное слово $w_I$.

$$\max y_{j^*} = \max \log y_{j^*} = \sum_{c=1}^{C} u_{j_c^*} - C \log \sum_{j'=1}^{V} \exp(u_{j'}) = \max p(w_{O,1}, w_{O,2}, ..., w_{O,C}|w_I)$$

Из формулы Байеса

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

понятно, что увеличение $p(w_{O,1}, w_{O,2}, ..., w_{O,C}|w_I)$ повлечет за собой увеличение $p(w_I|w_{O,1}, w_{O,2}, ..., w_{O,C})$.

В вышеописанных моделях вычисление функции активации софтмакс линейно зависит по времени от объема словаря $V$. В реальных задачах $V$ может достигать

нескольких сотен тысяч, тогда использование данной функции активации очень затратно по времени, поэтому в реализации Word2Vec используются метод для быстрого вычисления функции активаци: иерархический софтмакс (Hierarchical Softmax).

В иерархическом софтмаксе по всем словам в словаре строится дерево Хафмана. В полученном дереве $V$ висячих вершин (листьев). Условная вероятность в этом случае вычисляется следующим образом:

$$p(v|w) = \prod_{n \in path(v)} \sigma(ch(n) w'_n h)$$

$\sigma(\cdot)-$сигмоидная функция активации.

$path(v)-$путь от вершины до корня.

$$ch(n) = \begin{cases} 1, & \text{если вершина n правый сын} \\ -1, & \text{если вершина n левый сын} \end{cases}$$

Оценим вычислительную сложность обучения моделей из Word2Vec. Сложность модели Continuous Bag of Words $Q_1 = N \times D + D \times \log_2(V)$. Сложность модели Skipgram $Q_2 = C \times (D + D \times log2(V))$.

$C-$ максимальное расстояние между словами в одном окне.

$V-$ число слов в словаре.

$N-$ число слов в обучающих данных.

$D-$ число нейронов на скрытом уровне.

Skipgram модель работает медленнее, но обычно с помощью нее достигается лучшее качество классификации текстов.

### 5.3.3 GloVe

GloVe[9] — технология разработанная в Стэнфордском университете. Позволяет для каждого слова в текстовых данных получить соответствующий вектор фиксированной длины с помощью статистической информации об этом слове в данных.

Пусть объем словаря данных равен V. Все слова встретившиеся в данных нумеруются от 1 до $V$ Составляется матрица слово-слово $X \in \mathbb{R}^{V \times V}$, где $x_{ij}$ - количество раз, когда слова слово i встречается в контексте слова j. Слово $a$ встречается в контексте слова $b$ если существует часть текста, где между ними не более девяти слов.

Обозначим $X_i = \sum_k x_{ik}$ (сумма i-ой строки). Тогда вероятность того что слово j встретилось в контексте слова i равна $P_{ij} = P(j|i) = \dfrac{x_{ij}}{X_i}$.

Заметим, что если слово $i$ встречается в контексте слова $k$ чаще чем слово $j$ встречается в контексте слова $k$, то $\dfrac{P_{ik}}{P_{jk}} > 1$, а $\dfrac{P_{jk}}{P_{ik}} < 1$

Хотим построить такую функцию $F(w_i, w_j, \widehat{w}_k)$, чтобы она показывала какое из слов $i$ или $j$ более вероятно встретиться в контексте слова $k$. Где $w_i, w_j, \widehat{w}_k$−векторные представления слов $i$, $j$ и $k$ соответственно.

$$F(w_i, w_j, \widehat{w}_k) = \frac{P_{ik}}{P_{jk}}$$

Авторы метода GloVe предлагают использовать

$$F((w_i - w_j)^T \widehat{w}_k) = \frac{F(w_i^T \widehat{w}_k)}{F(w_j^T \widehat{w}_k)}, \text{ где } F(w_i^T \widehat{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

Тогда в качестве функции $F(\cdot)$ можно выбрать $F(x) = \exp(x)$, а вектор $w_i$ взять таким, чтобы

$$w_i^T \widehat{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

Теперь, учитывая, что $\log(X_i)$ фиксирован, перепишем задачу следующим образом

$$w_i^T \widehat{w}_k + b_i + \widehat{b}_k = \log(X_{ik}),$$
$$\text{где } b_i + \widehat{b}_k = \log(X_i)$$

В итоге авторы используют функцию потерь $J$ и настраивают модель с помощью алгоритма *AdaGrad* [4]

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widehat{w}_j + b_i + \widehat{b}_j - \log X_{ij})^2$$

Функция $f(x)$ должна удовлетворять следующим условиям: $f(0) = 0$; $f(x)$−не убывает; $f(x)$− относительно маленькая для больших значений $x$.

Авторы метода использовали

$$f(x) = \begin{cases} (x/x_{max})^\alpha, & x < x_{max} \\ 1, & \text{иначе} \end{cases}$$

Параметры были выбраны эмпирическим путем:

$$\alpha = 3/4$$
$$x_{max} = 100$$

# 6 Эксперименты

## 6.1 Данные

Эксперименты проводились на данных из статьи [13], в которой приведены результаты тестирования сверточной нейронной сети с посимвольным подходом, а так же многих традиционных методов классификации текстов. Ниже приведена информация об используемых текстах.

Таблица 1: Данные

| Выборка | Число классов | Размер обучающей выборки | Размер тестовой выборки |
|---|---|---|---|
| Ag news | 4 | 120000 | 7600 |
| DBPedia | 14 | 560000 | 70000 |
| Amazon Review Full | 5 | 3000000 | 650000 |

1. Ag news — новостные интернет статьи . Объем обучающей выборки 120000 объектов, объем тестовой выборки 7600 объектов. Статьи необходимо классифицировать на 4 класса — мировые, спортивные, бизнес и научные новости.

2. DBPedia — название и аннотации статей из Википедии. Объем обучающей выборки 560000 объектов, объем тестовой выборки 70000 объектов. Тексты необходимо классифицировать на 14 классов — компания, образовательное учреждение, художник, спортсмен, чиновник, средство передвижения, здание, природное место, деревня, животное, завод, альбом, фильм, литературное произведение.

3. Amazon Review Full — комментарии с сайта Amazon.com. Объем обучающей выборки 3000000 объектов, объем тестовой выборки 600000 объектов. Тексты необходимо классифицировать на 5 классов — отзывы пользователей от отрицательного до положительного по пятибалльной шкале.

## 6.2 Посимвольный подход

Реализована сверточная нейронная сеть с посимвольным подходом для классификации текстов, описанная в разделе 5.2.1. В данном подходе использовался алфавит из символа перевода строки и следующих 69 символов:

$abcdefghijklmnopqrstuvwxyz$0123456789−, ; .!? :′ ”/\|_@#$%ˆ&∗˜‘+− =<> ()[]{}

Из каждого объекта выбраны первые 1014 символов и далее только они учитываются при классификации. Данные символы переводятся в матрицу размера $70 \times 1014$, а затем подаются на вход сверточной нейронной сети. Все буквы английского алфавита в тексте приводятся к нижнему регистру. Веса нейронной сети инициализируются из нормального распределения $\mathcal{N}(0, 0.05)$.

Архитектура сверточной нейронной сети описана в Таблице 2. Между полносвязными слоями 6, 7 и 7, 8 использовалась dropout регуляризация с параметром $p = 0.5$. При тестировании сети параметр $p = 1.0$. Функция активации на всех слоях кроме последнего — Relu, на последнем — Softmax. Нейронноя сеть минимизировала кросс-энтропийную функцию потерь. Данная архитектура описана в статье [13]

Таблица 2: Архитектура сверточной нейронной сети

| № слоя | Количество карт признаков | Размер ядра свертки | Размер ядра max-pooling |
|---|---|---|---|
| 1 | 256 | 7 | 3 |
| 2 | 256 | 7 | 3 |
| 3 | 256 | 3 | — |
| 4 | 256 | 3 | — |
| 5 | 256 | 3 | — |
| 6 | 256 | 3 | 3 |
| 7 | 1024 | — | — |
| 8 | 1024 | — | — |
| 9 | число классов | — | — |

Нейронная сеть реализована с использованием библиотеки Tensor flow. Обучение и тестирование происходило на арендованной машине Amazon G2 Instances с одной видеокартой Nvidia Kepler GK104 и 8 ядерным процессором Intel Xeon E5-2670. Это позволило значительно ускорить выполнение программы.

Так как обучение на всем количестве объектов не представляется возможным из-за ограничения памяти на видеокарте, то обучение производилось в несколько итераций на каждой из которых выбирались случайные 140 объектов, которые подавались на вход нейронной сети.

## 6.3 Предобработка текста

В реализованной сверточной нейронной сети используется лишь первые 1014 символов из каждого документа. Для достижения лучшего качества классификации необходимо, чтобы в используемых символах содержалось как можно больше информации о классовой принадлежности документа.

В выбранной части текста могут присутствовать шумовые слова, которые не несут информации о данном тексте, если их удалить, то, возможно, удастся использовать больше информативных символов текста. Если в тексте встречаются одинаковые слова, но с разными окончаниями, при помощи стемминга и лемматизации можно их привести к одному виду и сократить общий объем каждого слова, поэтому при их использовании так же, возможно, удастся достичь лучшего качества классификации.

В данной работе проводилось исследование, как влияет предобработка текста на качество классификации с помощью сверточных нейронных сетей с посимвольным подходом. Использовалось несколько способов предобработки

1. Стэмминг — удаление окончаний, приведение слова к основе.
   Реализовано с помощью пакета nltk.stem.

2. Лемматизация — приведение слова к начальной форме.
   Реализовано с помощью пакета nltk.stem.

3. Удаление стоп-слов из списка nltk.corpus.

## 6.4 Результаты

Результаты классификации оценивались с помощью метрики accuracy, т. е. считалась доля верно классифицированных объектов к общему количеству объектов.

Ниже в таблице приведены итоги тестирования описанной сверточной нейронной сети на различных данных и качество классификации этих же данных, полученное в статье [13] с помощью методов *Bag of Words*, *Bag of Words & TFIDF*, *Bag of Ngrams & TFIDF*, а так же качество классификации аналогичной сверточной нейронной сети с посимвольным подходом из статьи [13] и сверточной нейронной сети с использованием кодирования слов и Word2Vec.

В Таблице 3 приведены результаты тестирования полученные экспериментально без предобработки текста и приведенные в статье [13]. Красным цветом в таблице выделено худшее качество классификации из приведенных методов, синим — лучшее.

Таблица 3: Accuracy, полученные экспериментально и приведенные в статье [13].

| Экспериментальные результаты | | | Результаты из статьи | | | | |
|---|---|---|---|---|---|---|---|
| Данные | Итерации | Accuracy | BoW | BoW T. | Ng. T | W2V | Conv. |
| Ag news | 5000 | 0.829 | 0.888 | 0.896 | 0.923 | 0.886 | 0.843 |
| DBPedia | 8000 | 0.953 | 0.966 | 0.973 | 0.986 | 0.982 | 0.980 |
| Amazon Rev. | 30000 | 0.563 | 0.546 | 0.552 | 0.524 | 0.574 | 0.594 |

Обозначения, использованные в Таблице 3

- BoW — Bag of Words (мешок слов) описано в разделе 3.2.1.
- BoW T. — Bag of Words & TFIDF описано в разделе 3.2.2.
- Ng. T. — Bag of Ngrams & TFIDF описано в разделе 3.2.3.
- W2V — Сверточная нейронная сеть с кодированием слов и Word2Vec.
- Conv. — Сверточная нейронная сеть с посимвольным подходом.

Из Таблицы 3 видно, что экспериментально не удалось повторить результаты сверточной нейронной сети описанной в статье [13]. Но удалось достичь близкого качества классификации на всех трех выборках.

Ниже на Рис. 11 и в Таблице 4 приведены результаты работы сверточной нейронной сети с посимвольным подходом с предобработкой текста.

Таблица 4: Accuracy, полученные без предобработки и с предобработкой.

| Данные | Итерации | Без предобработки | Стоп-слова | Стэмминг | Лемматизация |
|---|---|---|---|---|---|
| Ag news | 5000 | 0.829 | 0.832 | 0.830 | 0.830 |
| DBPedia | 8000 | 0.953 | 0.949 | 0.944 | 0.943 |
| Amz Rev | 30000 | 0.563 | 0.558 | 0.554 | 0.555 |

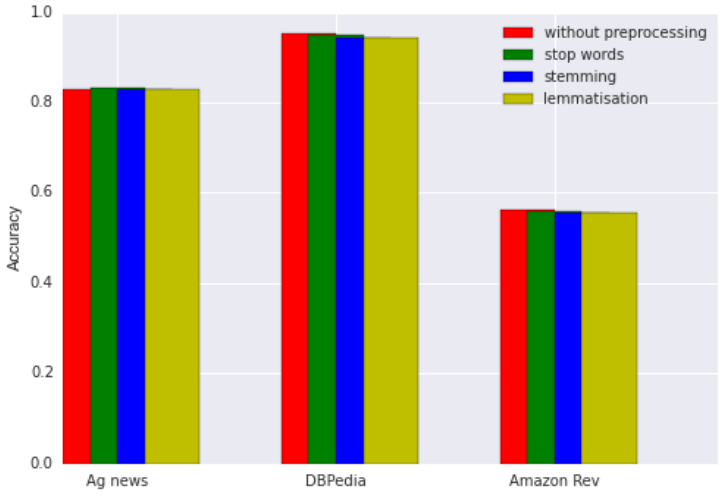


Рис. 11: Результаты работы сверточной нейронной сети по метрике Accuracy с посимвольным подходом с предобработкой текста

Из графика видно, что на используемых данных удаление стоп-слов и применение стэмминга и лемматизации почти не повлияло на изменение качества классификации. Причем на данных Ag news удаление стоп слов позволило незначительно улучшить результат по метрике Accuracy, а на выборках большего размера улучшить качество с помощью предобработки не удалось.

## 6.5 Выводы

На основании результатов проведенных экспериментов реализованной сверточной нейронной сети и на основании статьи [7] можно сделать следующие выводы:

- Сверточные нейронные сети с посимвольным подходом — эффективный метод для классификации текстов. Наиболее важный вывод из проведенных экспериментов, что данный метод может классифицировать тексты с высокой точностью без использования слов. Это значит, что естественный язык можно рассматривать как сигнал.

- Посимвольный подход позволяет решать задачу классификации текстов с нуля, т. е. не нужны никакие знания о синтаксической или семантической структуре языка, чтобы с хорошей точностью понимать о чем текст.

- На небольших наборах (до нескольких 100 тысяч) данных лучше работают традиционные методы. Когда данных становится больше (более 1 миллиона текстов), лучше работают сверточные нейронные сети с посимвольным подходом. Это еще раз подтверждает, что нет ни одного алгоритма машинного обучения, который может работать на всех видах наборов данных.

- При использовании предобработки текста на выбранных данных не удалось значительно улучшить качество классификации. Возможно, это связано с тем, что тестирование производилось на выборках с небольшой длиной текста в каждом документе.

# 7 Заключение

В данной работе проводилось исследование основных методов классификации текстов, в том числе и методов классификации текстов с использованием нейронных сетей.

Нейронные сети, зарекомендовавшие себя, как мощный алгоритм для классификации изображений, в последнее время стали активно использоваться и для других задач машинного обучения.

В данной работе было рассмотренно два основных метода использования сверточных нейронных сетей для задачи классификации текста: посимвольный подход и подход с использованием кодирования слов. Реализована сверточная нейронная сеть с посимвольным подходом, работа которой проверена на трех выборках разного размера. Произведено сравнение качества классификации на этих же данных с другими методами. Показано, что данная нейронная сеть справляется с задачей классификации текстов иногда лучше, чем все остальные рассмотренные методы. Исследовано влияние на качество классификации некоторых алгоритмов предобработки текста.

# Список литературы


[1] Damashek, M. Gauging similarity with n-grams: Language-independent categorization of text / Marc Damashek // Science, New Series. — 1995.

[2] Efficient estimation of word representations in vector space / Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean // ICLR. — 2013.

[3] Harris, Z. Distributional structure / Zellig Harris // Word. — 1954.

[4] John Duchi Elad Hazan, Y. S. Adaptive subgradient methods for online learning and stochastic optimization / Yoram Singer John Duchi, Elad Hazan // JMLR. — 2011.

[5] Kim, Y. Convolutional neural networks for sentence classification / Yoon Kim // IEMNLP. — 2014. — Sep. — 1746 -1751 p.

[6] Krizhevsky, A. Imagenet classification with deep convolutional neural networks / Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton // NIPS. — 2012. — 1106 -1114 p.

[7] LeCun, X. Z. Y. Text understanding from scratch / Xiang Zhang Yann LeCun // Computer Science Department. — 2016.

[8] McCulloch, W. S. A logical calculus of the ideas immanent in nervous activity / Warren S. McCulloch, Walter Pitts // Springer New York. — 1943.

[9] Pennington, J. Glove: Global vectors for word representation / Jeffrey Pennington, Richard Socher, Christopher D // EMNLP. — 2014. — 1532 -1543 p.

[10] S, J. K. A statistical interpretation of term specificity and its application in retrieval / Jones K. S // Journal of Documentation. — 1972.

[11] X, R. word2vec parameter learning explained / Rong X // arXiv:1411.2738. — 2014.

[12] Yann LeCun Leon Bottou, Y. B. Gradient-based learning applied to document recognition / Yoshua Bengio Yann LeCun, Leon Bottou, Patrick Haffner // IEEE. — 1998.

[13] Zhang, X. Character-level convolutional networks for text classification / Xiang Zhang, Junbo Zhao, Yann LeCun // In Advances in Neural Information Processing Systems. — 2015. — Feb. — 649 - 657 p.

[14] Воронцов, К. В. Курс лекций по машиному обучению / К. В. Воронцов. — 2015.