Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

# МАШИННОЕ ОБУЧЕНИЕ

## ЛАБОРАТОРНАЯ РАБОТА №7
«Рекуррентные нейронные сети для анализа текста»

БГУИР 1-40 81 04

Магистрант:
гр. 858641
Кукареко А.В.

Проверил:
Стержанов М. В.

Минск, 2020

## ХОД РАБОТЫ

**Данные**.

Набор данных для предсказания оценок для отзывов, собранных с сайта imdb.com, который состоит из 50,000 отзывов в виде текстовых файлов. Отзывы разделены на положительные (25,000) и отрицательные (25,000). Данные предварительно токенизированы по принципу “мешка слов”, индексы слов можно взять из словаря (imdb.vocab). Обучающая выборка включает в себя 12,500 положительных и 12,500 отрицательных отзывов, контрольная выборка также содержит 12,500 положительных и 12,500 отрицательных отзывов, а также. Данные можно скачать по ссылке https://ai.stanford.edu/~amaas/data/sentiment/ .

**Задание.**

1. Загрузите данные. Преобразуйте текстовые файлы во внутренние структуры данных, которые используют индексы вместо слов;
2. Реализуйте и обучите двунаправленную рекуррентную сеть (LSTM или GRU). Какого качества классификации удалось достичь?
3. Используйте индексы слов и их различное внутреннее представление (word2vec, glove). Как влияет данное преобразование на качество классификации?
4. Поэкспериментируйте со структурой сети (добавьте больше рекуррентных, полносвязных или сверточных слоев). Как это повлияло на качество классификации?
5. Используйте предобученную рекуррентную нейронную сеть (например, DeepMoji или что-то подобное).

**Результат выполнения:**

1. Загрузите данные. Преобразуйте текстовые файлы во внутренние структуры данных, которые используют индексы вместо слов.

Оригинальный словарь «imdb.vocab» состоит из 89527. Из этого словаря было решено взять 10 000 наиболее частых.

Все файлы с отзывами были сконвертированы в «DataFrame», пример можно увидеть на рисунке 1.

| | label | rank | id | text |
|---|---|---|---|---|
| 0 | 1 | 10 | 1926 | In 1967 I visited the Lake Elsinore glider-por... |
| 1 | 1 | 9 | 6677 | A film about wannabee's, never-were's and less... |
| 2 | 1 | 10 | 11812 | This movies is the best movie to watch for com... |
| 3 | 1 | 7 | 9567 | I watched the DVD of this movie which also com... |
| 4 | 1 | 9 | 4511 | I just picked up the DVD release of this movie... |

Рисунок 1 – оригинальные данные.

Затем текст в «датафрейме» был обработан специальным образом. Обработка включала в себя:

- удаление html из текста;
- удаление одиночных букв и слов длинной меньше 2х символов;
- удаление цифр;
- удаление знаков пунктуации;
- удаление «стоп-слов» (пакет nltk.corpus).

Результаты обработки можно увидеть на рисунке 2.

| | label | rank | id | text |
|---|---|---|---|---|
| 0 | 1 | 10 | 1926 | [visited, lake, elsinore, gliderport, flew, ye... |
| 1 | 1 | 9 | 6677 | [film, wannabees, neverweres, lessthanheroes, ... |
| 2 | 1 | 10 | 11812 | [movies, best, movie, watch, comic, book, feel... |
| 3 | 1 | 7 | 9567 | [watched, dvd, movie, also, comes, excellent, ... |
| 4 | 1 | 9 | 4511 | [picked, dvd, release, movie, holiday, norway,... |

Рисунок 2 – данные после обработки текста.

Финальная часть обработки заключалась в замене слов из отзывов на их частотные индексы из словаря «imdb.vocab». Результат можно увидеть на рисунке 3.

| | label | rank | id | text |
|---|---|---|---|---|
| 0 | 1 | 10 | 1926 | [5390, 2025, 2, 2, 8434, 4287, 2, 319, 2, 3480... |
| 1 | 1 | 9 | 6677 | [18, 2, 2, 2, 245, 3985, 542, 2, 2, 1380, 2, 2... |
| 2 | 1 | 10 | 11812 | [100, 116, 16, 104, 716, 264, 232, 706, 1282, ... |
| 3 | 1 | 7 | 9567 | [286, 279, 16, 80, 257, 310, 1639, 1392, 641, ... |
| 4 | 1 | 9 | 4511 | [1596, 279, 764, 16, 3134, 2, 616, 641, 2494, ... |

Рисунок 3 – слова заменены на индексы.

Классы являются сбалансированными. Это можно увидеть на рисунке 4.

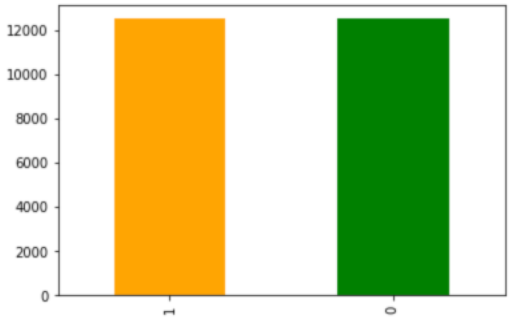


Рисунок 4 – баланс классов.

Далее был проведен анализ длинны отзывов. На рисунке 5 можно увидеть, что большинство отзывов имеют длину до 50 слов. Исходя из этой информации, размер вектора отзыва был выбран – 130.

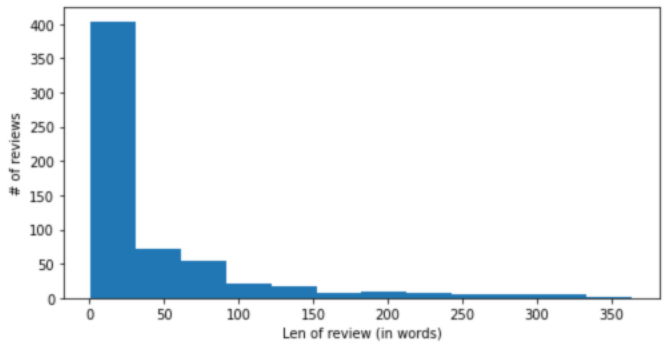


Рисунок 5 - кол-во изображений в каждом классе.

2. Реализуйте и обучите двунаправленную рекуррентную сеть (LSTM или GRU). Какого качества классификации удалось достичь?

В этом задании было реализовано 2 модели, первая с слоем LSTM, вторая с GRU. Архитектуру первой модели можно увидеть в таблице 1.

Таблица 1 – Архитектура нейронной сети с LSTM.

| **Слой** | | **Размер** | **Активация** |
|---|---|---|---|
| Входной | - | 130 | - |
| 1 | Embedding | 130x2 | - |
| 2 | Bidirectional-LSTM(64) | 128 | tanh |
| 3 | Dropout(0.5) | - | |
| Выходной | FC | 1 | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 5;
- batch size - 64.

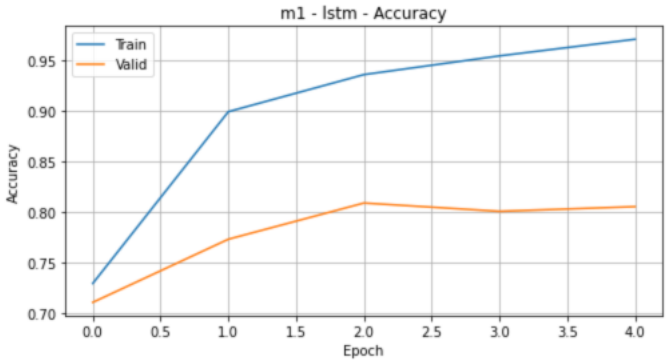


Рисунок 6 – график изменения accuracy модели (LSTM).

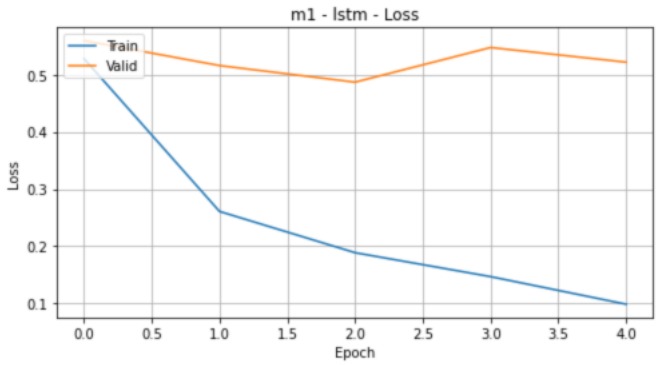


Рисунок 7 – график изменения loss модели (LSTM).

На тестовой выборке модель показала следующий результат:

- loss - 0.3604;
- accuracy – 0.8551.

Вторая модель имеет аналогичную архитектуру, за исключением того, что вместо слоя LSTM используется слой GRU. Архитектуру сети можно посмотреть в таблице 2.

Таблица 2 – Архитектура нейронной сети с GRU.

| **Слой** | | **Размер** | **Активация** |
|---|---|---|---|
| Входной | - | 130 | - |
| 1 | Embedding | 130x2 | - |
| 2 | Bidirectional-GRU(64) | 128 | tanh |
| 3 | Dropout(0.5) | - | |
| Выходной | FC | 1 | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 5;
- batch size - 64.

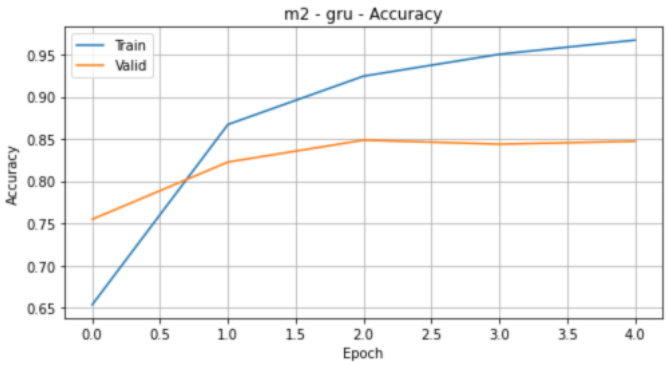


Рисунок 8 – график изменения accuracy модели (GRU).

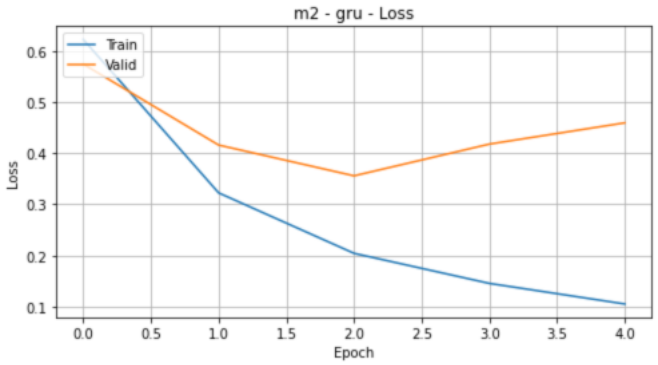


Рисунок 9 – график изменения loss модели (GRU).

На тестовой выборке модель показала следующий результат:

- loss - 0.3391;
- accuracy – 0.8539.

3. Используйте индексы слов и их различное внутреннее представление (word2vec, glove). Как влияет данное преобразование на качество классификации?

Для того, чтобы использовать индексы слов, мной был скачал словарь «glove.6B.100d.txt», который обучался на Wikipedia. Длинна вектора в словаре составляет 100 значений. Архитектуру новой сети можно посмотреть в таблице 3.

Таблица 3 – Архитектура нейронной сети с GloVe.

| **Слой** | | **Размер** | **Активация** |
|---|---|---|---|
| Входной | - | 130 | - |
| 1 | Embedding | 130x100 | - |
| 2 | Bidirectional-LSTM(64) | 128 | tanh |
| 3 | Dropout(0.5) | - | |
| Выходной | FC | 1 | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 8;
- batch size - 64.

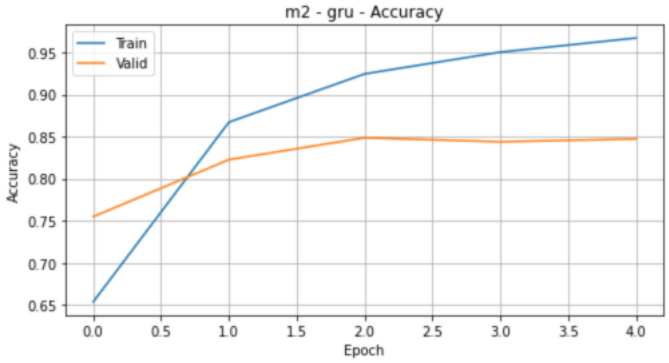


Рисунок 10 – график изменения accuracy модели (GloVe).

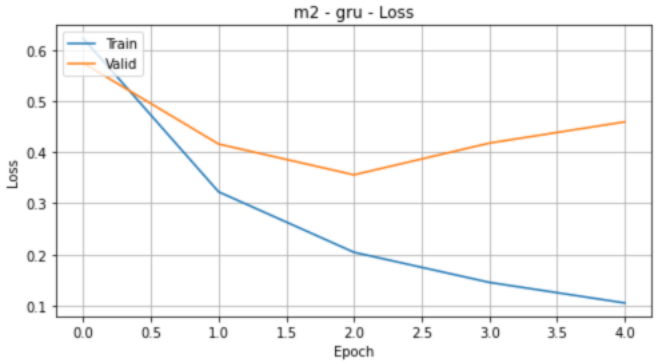


Рисунок 11 – график изменения loss модели (GloVe).

На тестовой выборке модель показала следующий результат:

- loss - 0.3567;
- accuracy – 0.8467.

Для использования Word2Vec была подключена модель из библиотеки «gensim.models.Word2Vec». Модель была обучена на всех словах из отзывов, с длинной выходного вектора – 100. Архитектуру новой сети можно посмотреть в таблице 4.

Таблица 4 – Архитектура нейронной сети с Word2Vec.

| **Слой** | | **Размер** | **Активация** |
|---|---|---|---|
| Входной | - | 130 | - |
| 1 | Embedding | 130x100 | - |
| 2 | Bidirectional-LSTM(64) | 128 | tanh |
| 3 | Dropout(0.5) | - | |
| Выходной | FC | 1 | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 8;
- batch size - 64.

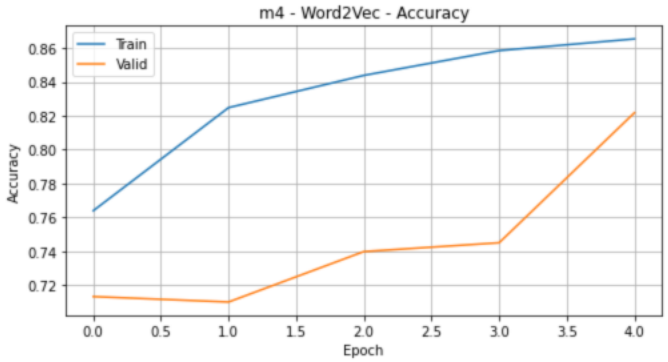


Рисунок 12 – график изменения accuracy модели (Word2Vec).

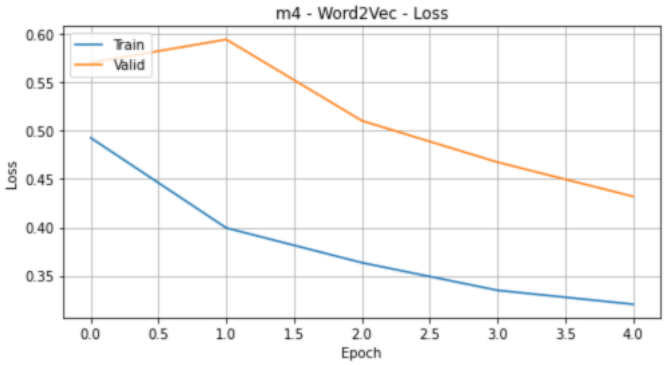


Рисунок 13 – график изменения loss модели (Word2Vec).

На тестовой выборке модель показала следующий результат:

- loss - 0.3392;
- accuracy - 0.8542.

4. Поэкспериментируйте со структурой сети (добавьте больше рекуррентных, полносвязных или сверточных слоев). Как это повлияло на качество классификации?

В качестве эксперимента, к прошлой модели использующей Word2Vec, был добавлен еще один слой LSTM. Архитектуру модели можно увидеть в таблице 5.

Таблица 5 – Архитектура нейронной сети с 2мя LSTM.

| **Слой** | | **Размер** | **Активация** |
|---|---|---|---|
| Входной | - | 130 | - |
| 1 | Embedding | 130x100 | - |
| 2 | Bidirectional-LSTM(64) | 128 | tanh |
| 3 | Bidirectional-LSTM(32) | 64 | |
| 4 | Dropout(0.5) | - | |
| Выходной | FC | 1 | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 5;
- batch size - 64.

На тестовой выборке модель показала следующий результат:

- loss - 0.3371;
- accuracy - 0.8502.

4. Используйте предобученную рекуррентную нейронную сеть (например, DeepMoji или что-то подобное).

Модель DeepMoji была установлена напрямую из репозитория - https://github.com/bfelbo/DeepMoji .

Особенность модели заключается в том, что она имеет механизм «Attention».

Attention представляет собой способ сообщить сети, на что стоит обратить больше внимания, то есть сообщить вероятность того или иного исхода в зависимости от состояния нейронов и поступающих на вход данных.

Тренировка нейросети с нуля (без загрузки пред обученных весов) была запущена со следующими параметрами:

- epochs – 3;
- batch size - 64.

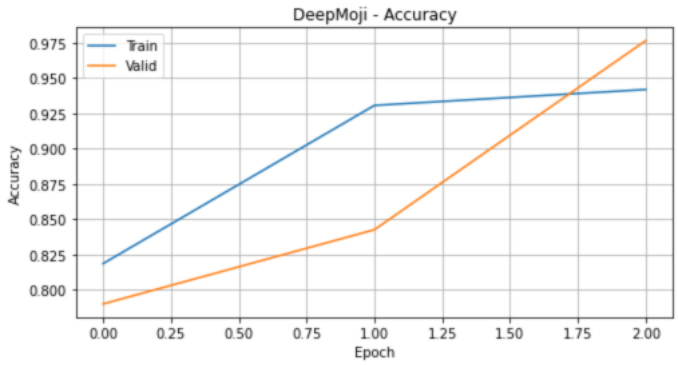


Рисунок 14 – график изменения accuracy модели (DeepMoji).

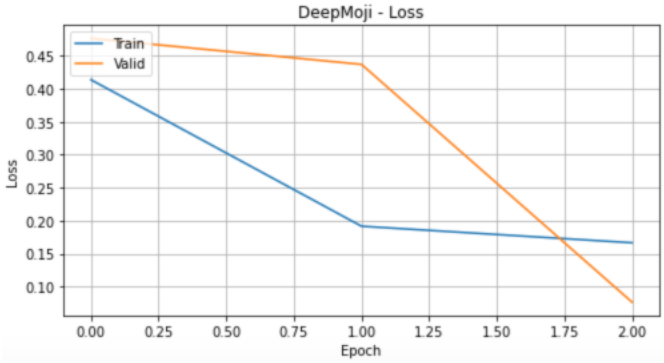


Рисунок 15 – график изменения loss модели (DeepMoji).

На тестовой выборке модель показала следующий результат:

- loss - 0.5502;
- accuracy - 0.8401.

DeepMoji с загруженными весами показала результат хуже:

- loss - 0.7140;
- accuracy - 0.5001.

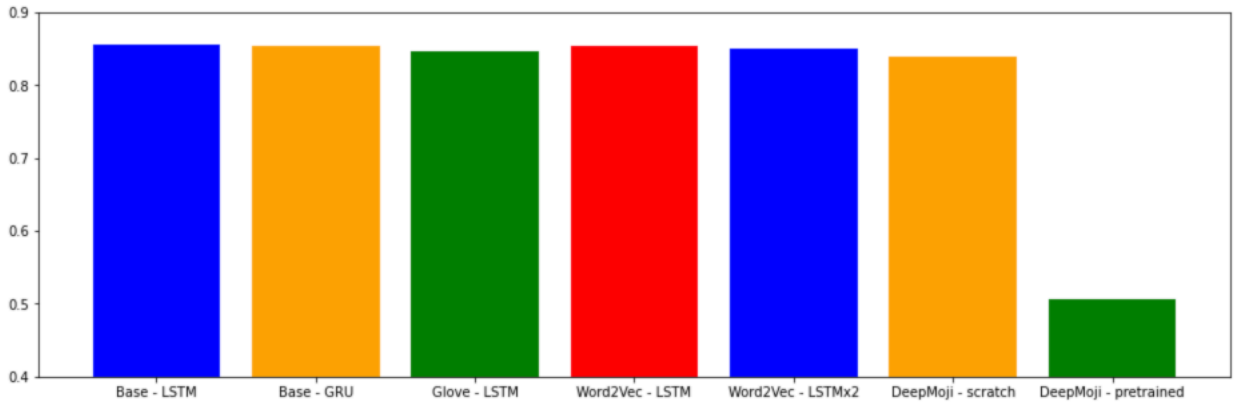


Рисунок 14 – сравнение accuracy моделей.

Как видно из рисунка 14, почти все модели выдают качество в рамках диапазона 84% - 86%. Применение векторных репрезентаций слов таких как GloVe и Word2Vec не помогли увеличить качество на данном наборе данных. Для повышения точности модели нужно более тщательно обработать текст, и поэкспериментировать с более сложными моделями.

Так же стоит отметить, что абсолютно все обученные модели страдали от «переобучения» даже не смотря на малое количество эпох (3-8).

**Вывод.**

В ходе выполнения лабораторной работы я построил несколько моделей использующих рекуррентные слои, такие как LSTM и GRU. Так же для векторного представления слов я использовал готовые решения GloVe и Word2Vec. В финальной части работы я применил готовую предобученную сеть «DeepMoji».

После обучения всех моделей и анализа результатов, можно сделать вывод, добиться каких-то «хороших» (точность 80% +) результатов в семантическом анализе текста довольно просто не требует больших вычислительных мощностей. Однако для достижения очень хороших результатов (точность 90% +) нужно приложить значительно больше усилий и времени.