ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Математическая кибернетика и информационные технологии» Дисциплина «Интеллектуальные системы»

Лабораторная работа № 8

«Генерация текста на основе “Алисы в стране чудес”»

Выполнил: \_\_\_\_\_\_\_ Д.С. Калининский

Проверил: \_\_\_\_\_\_\_ А.В. Шереметьев

Москва 2022

**Оглавление**

[1 Цель работы 3](#_Toc119580633)

[2 Задачи 3](#_Toc119580634)

[3 Выполнение работы 3](#_Toc119580635)

[4 Результаты 5](#_Toc119580636)

[5 Вывод 7](#_Toc119580637)

# **1 Цель работы**

Рекуррентные нейронные сети также могут быть использованы в качестве генеративных моделей.

Это означает, что в дополнение к тому, что они используются для прогнозных моделей (создания прогнозов), они могут изучать последовательности проблемы, а затем генерировать совершенно новые вероятные последовательности для проблемной области.

Подобные генеративные модели полезны не только для изучения того, насколько хорошо модель выявила проблему, но и для того, чтобы узнать больше о самой проблемной области.

# **2 Задачи**

1) Ознакомиться с генерацией текста;

2) Ознакомиться с системой Callback в Keras.

# **3 Выполнение работы**

Начнем с импорта необходимых зависимостей для предварительной обработки данных и построения модели (рисунок 1).

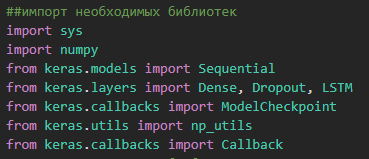


Рисунок 1 – Импорт библиотек

Затем нам нужно загрузить текст ASCII для книги в память и преобразовать все символы в нижний регистр, чтобы уменьшить словарный запас, который должна выучить сеть (рисунок 2).



Рисунок 2 – Загрузка текста

Теперь, когда книга загружена, мы должны подготовить данные для моделирования нейронной сетью. Мы не можем моделировать символы напрямую, вместо этого мы должны преобразовать символы в целые числа.

Мы можем сделать это легко, сначала создав набор всех отдельных символов в книге, а затем создав карту каждого символа с уникальным целым числом (рисунок 3).



Рисунок 3 – Преобразование символов в целые числа

Теперь, когда книга загружена и карта подготовлена, мы можем суммировать набор данных (рисунок 4).



Рисунок 4 – Суммируем набор данных

Мы видим, что книга содержит менее 150000 символов и что при преобразовании в строчные буквы в словаре есть только 47 различных символов для изучения сетью.

Гораздо больше, чем 26 в алфавите.

Теперь нам нужно определить данные обучения для сети Существует большая гибкость в том, как вы решаете разбивать текст и выставлять его в сети во время обучения. В этом уроке мы разделим текст книги на подпоследовательности с фиксированной длиной в 100 символов произвольной длины. Мы могли бы так же легко разделить данные по предложениям, дополнить более короткие последовательности и укоротить более длинные.

Каждый обучающий шаблон сети состоит из 100 временных шагов одного символа (X), за которыми следует один символьный вывод (y). При создании этих последовательностей мы перемещаем это окно по всей книге по одному символу за раз, позволяя каждому персонажу выучить шанс из 100 предшествующих ему символов (кроме, конечно, первых 100 символов).

Например, если длина последовательности равна 5 (для простоты), то первые два шаблона обучения будут следующими:

CHAPT -> E

HAPTE -> R

Разделяя книгу на эти последовательности, мы конвертируем символы в целые числа, используя нашу таблицу поиска, которую мы подготовили ранее (рисунок 5).

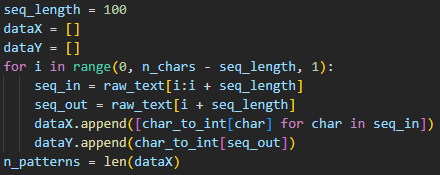


Рисунок 5 – Разделяем книгу на последовательности

Теперь, когда мы подготовили наши тренировочные данные, нам нужно преобразовать их так, чтобы они подходили для использования с Keras.

Сначала мы должны преобразовать список входных последовательностей в форму [образцы, временные шаги, особенности] ожидается сетью LSTM.

Затем нам нужно изменить масштаб целых чисел в диапазоне от 0 до 1, чтобы облегчить изучение шаблонов сетью LSTM, которая по умолчанию использует функцию активации сигмовидной кишки.

Наконец, нам нужно преобразовать выходные шаблоны (отдельные символы, преобразованные в целые числа) в одну кодировку. Это сделано для того, чтобы мы могли настроить сеть так, чтобы она предсказывала вероятность каждого из 47 различных символов в словаре (более простое представление), а не пыталась заставить ее предсказать точно следующий символ. Каждое значение y преобразуется в разреженный вектор длиной 47, полный нулей, за исключением 1 в столбце для буквы (целое число), которую представляет шаблон.

Мы можем реализовать эти шаги, как показано ниже (рисунок 6).



Рисунок 6 – Реализация шагов

Теперь мы можем определить нашу модель LSTM. Здесь мы определяем один скрытый слой LSTM с 256 единицами памяти. Сеть использует выпадение с вероятностью 20.

Выходной уровень — это Плотный уровень, использующий функцию активации softmax для вывода прогнозирования вероятности для каждого из 47 символов в диапазоне от 0 до 1.

Эта проблема на самом деле представляет собой проблему классификации отдельных символов с 47 классами, и поэтому она определяется как оптимизация потерь (перекрестная энтропия) с использованием алгоритма оптимизации ADAM по скорости (рисунок 7).

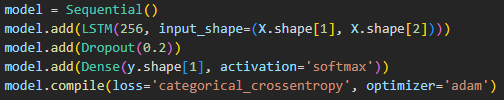


Рисунок 7 – Определяем модель

Тестового набора данных нет. Мы моделируем весь обучающий набор данных, чтобы узнать вероятность каждого символа в последовательности.

Сеть работает медленно. Из-за медлительности и из-за наших требований по оптимизации мы будем использовать контрольные точки модели для записи всех сетевых весов, чтобы каждый раз регистрировать улучшение потерь в конце эпохи. Мы будем использовать лучший набор весов (наименьшая потеря), чтобы реализовать нашу генеративную модель в следующем разделе (рисунок 8).

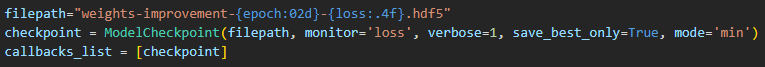


Рисунок 8 – Использование контр. точек

Теперь осталось обучить модель (рисунок 9).



Рисунок 9 – Обучение модели

Результат запуска программы представлен на рисунке 10.

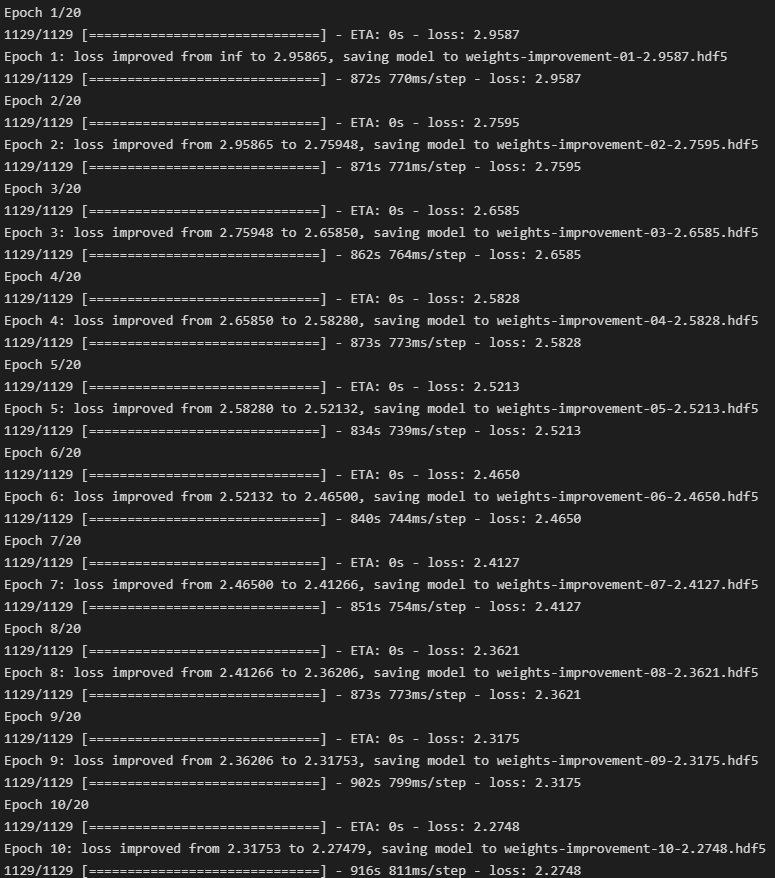


Рисунок 10 - Результат

Генерация текста с использованием обученной сети LSTM относительно проста. Во-первых, мы загружаем данные и определяем сеть точно таким же образом, за исключением того, что веса сети загружаются из файла контрольных точек, и сеть не нуждается в обучении (рисунок 11).

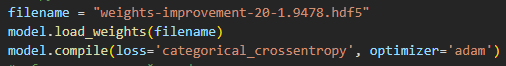


Рисунок 11 – Загружаем данные и определяем сеть

Кроме того, при подготовке сопоставления уникальных символов с целыми числами мы также должны создать обратное отображение, которое мы можем использовать для преобразования целых чисел обратно в символы, чтобы мы могли понять предсказания.

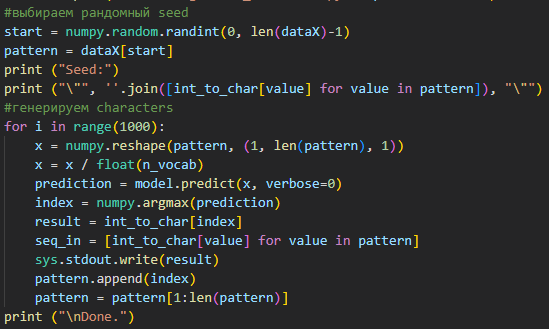


Рисунок 12 – Генерируем seed

Простейший способ использования модели Keras LSTM для прогнозирования – сначала начать с последовательности начальных чисел в качестве входных данных, сгенерировать следующий символ, затем обновить последовательность начальных чисел, чтобы добавить сгенерированный символ в конце, и обрезать первый символ. Этот процесс повторяется до тех пор, пока мы хотим предсказать новые символы (например, последовательность длиной 1000 символов).

Мы можем выбрать случайный шаблон ввода в качестве нашей начальной последовательности, а затем распечатать сгенерированные символы по мере их генерации (рисунок 13).

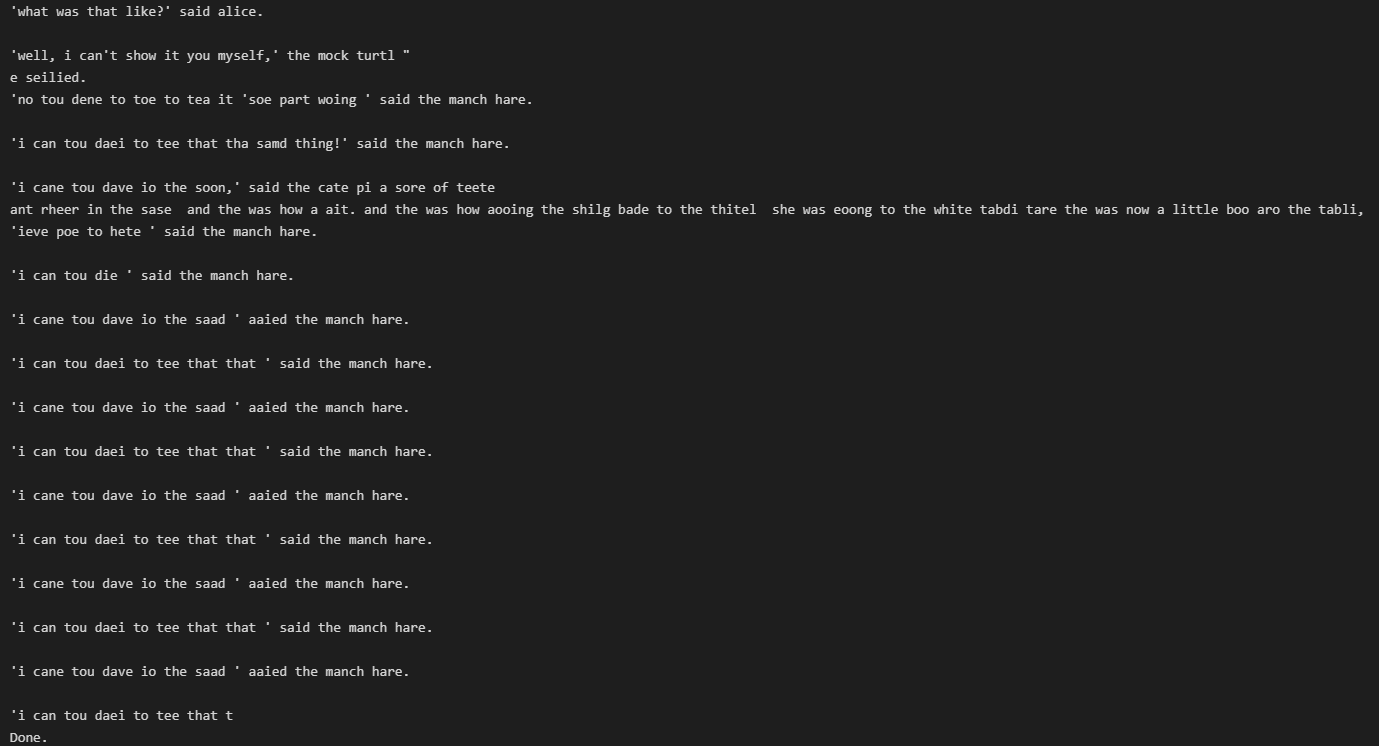


Рисунок 13 - Результат

# **5 Вывод**

В данной лабораторной работе, была использована рекуррентная нейронная сеть в качестве генеративных моделей. Также ознакомились с системой Callback в Keras.