УДК 004.934.1 **Информационные технологии**

Вешкин Игорь Ильич, *студент-магистрант 2 курса,*
факультета математики и информационных технологий
ФГБОУ ВО «НИ МГУ им. Н.П. Огарёва», г. Саранск

ПРИНЦИПЫ ПРЕДСТАВЛЕНИЯ АУДИО – ФАЙЛОВ ПРИ ПРОЕКТИРОВАНИИ НЕЙРОННЫХ СЕТЕЙ РАСПОЗНАВАНИЯ РЕЧИ

Аннотация: Данная статья посвящена рассмотрению способа представления аудио-файлов в нейронных сетях распознавания речи и их преобразования на разных этапах работы сети. В рамках изложенного материала были рассмотрены как основные форматы аудио-файлов вместе с их отличиями, так и их преобразования в памяти компьютера.

Ключевые слова: Нейронные сети, свёрточные нейронные сети, спектрограммы, аудио-файлы, нормализация, MP3, FLAC, WAV.

Abstract: This article is devoted to the consideration of the way audio files are represented in neural speech recognition networks and their transformation at different stages of the network. Within the framework of the presented material, both the main audio file formats, along with their differences, and their transformations in computer memory were considered.

Keywords: Neural networks, convolutional neural networks, spectrograms, audio files, normalization, MP3, FLAC, WAV.

Введение

Нейронные сети распознавания речи постепенно становятся важной частью жизни современного человека. Они используются при работе голосовых помощников, построенные автоматических субтитров сервисов потоковой трансляции видео, формирование запросов поисковых систем и переводчиков и

так далее. На фоне популярности данных технологий, у многих специалистов, начавших изучать нейронные сети, возникает вопросы о способах представления аудио-файлов в форме, которую сможет воспринять модель распознавания речи.

Принципы представления аудио-файлов в Machine Learning

Важным вопросом при разработке нейронной сети распознавания речи является выбор формы представления аудио-файлов, которые сможет обработать сеть. Для начала важно рассмотреть основные особенности наиболее распространенных аудио-форматов (WAV, FLAC, MP3).

Формат MP3 является наиболее распространенным в повседневном использовании, но в связи с применением алгоритмов сжатия аудио-дорожки, с целью экономии места жесткого диска, плохо подходит для хранения аудио записей, использующихся при обучении нейронной сети, поскольку имеет место потеря качества [1].

Формат FLAC также использует сжатие, но в отличие от MP3, потери качества аудио-дорожки более низкие, поэтому используются при обучении [1].

Формат WAV является эталонным, поскольку не использует сжатие. WAV является CD-стандартом, который используется при дистрибуции музыки на физических носителях. Единственный недостаток данного формата – это гораздо больший объем, занимаемый одним файлом. При использовании wav-формата аудио-файлов в dataset-ax, время требуемое для обработки такого массива данных увеличивается, поскольку в полноценных наборах данных количество аудио-файлов может доходить до нескольких сотен тысяч, которые никаким образом не были сжаты [1].

Рассмотрим преобразование аудио-файлов в форму, который сможет воспринять нейронная сеть, на примере набора данных «mini speech commands», в котором используются файлы wav-расширения. В данном dataset-е всего 8 команд: left, go, no, up, right, down, stop, yes; под каждую команду отведено ровно 1000 аудио-записей, каждая из которых была записана отдельным человеком. Также аудио-файлы не были записаны в идеальных условиях, что несомненно повысит качество работы сети с аудио, в которых присутствуют сторонние шумы

и другие недостатки записывающей аппаратуры [2].

Изначально на вход поступает файл wav-формата, который декодируется в тензор, многомерный массив данных, каждый элемент которого нормализован и приведен к типу float32 [2].

Нормализация – это процесс приведения больших числовых данных к их эквивалентным значениям, удовлетворяющих определенному числовому промежутку, как правило [-1; 1]. Пример подобного преобразования приведен на рисунках 1 и 2.

$$OriginalArray = [-20,\ -8,\ 0,\ 7,\ 4,\ -2,\ 20,\ 23,\ 15,\ -6,\ 18,\ 6]$$

Рисунок 1 – Массив данных до нормализации

$$NormalizedArray = [-1.0,\ -0.4419,\ -0.0698,\ 0.2558,\ 0.1163,\ -0.1628,\ 0.8605,\ 1.0,\ 0.6279,\ -0.3488,\ 0.7674,\ 0.2093]$$

Рисунок 2 – Массив данных после нормализации

Нормализация требуется для оптимизации работы ПК с исходными данными, поскольку выполнение алгебраических операций над элементами меньшего значения производится быстрее и занимает меньше места памяти компьютера [3].

Затем на базе имеющегося тензора формируется спектрограмма, изображение описывающие зависимость между силой звукового сигнала и времени [2]. Пример спектрограммы приведен на рисунке 3.

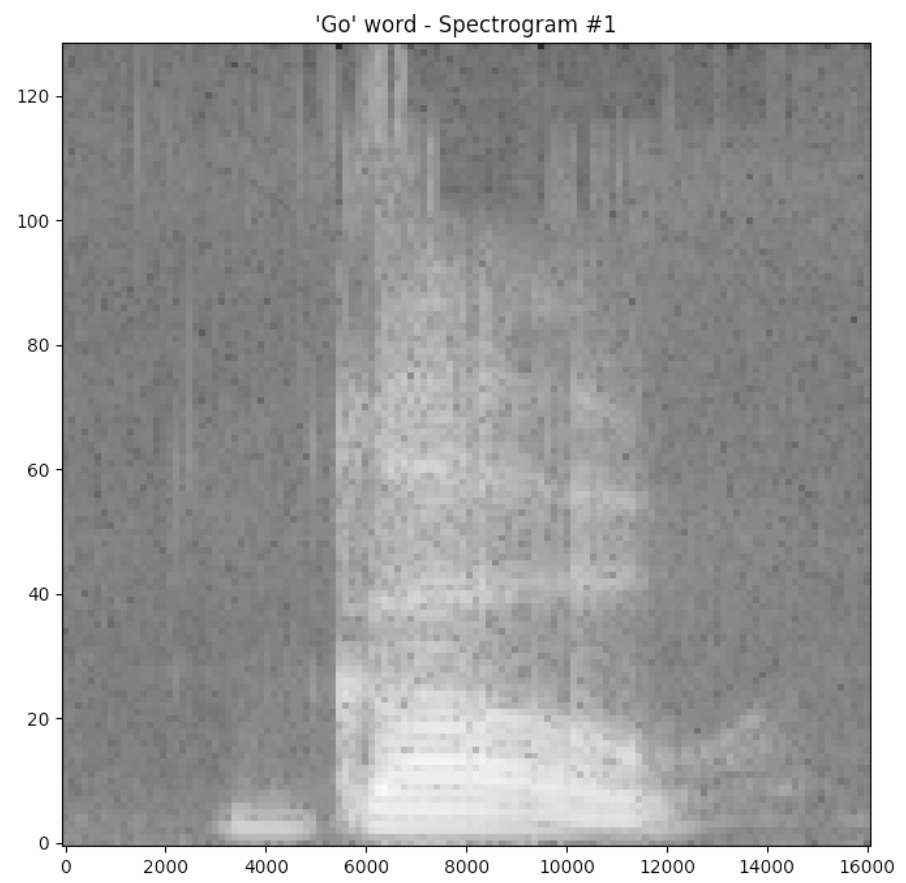
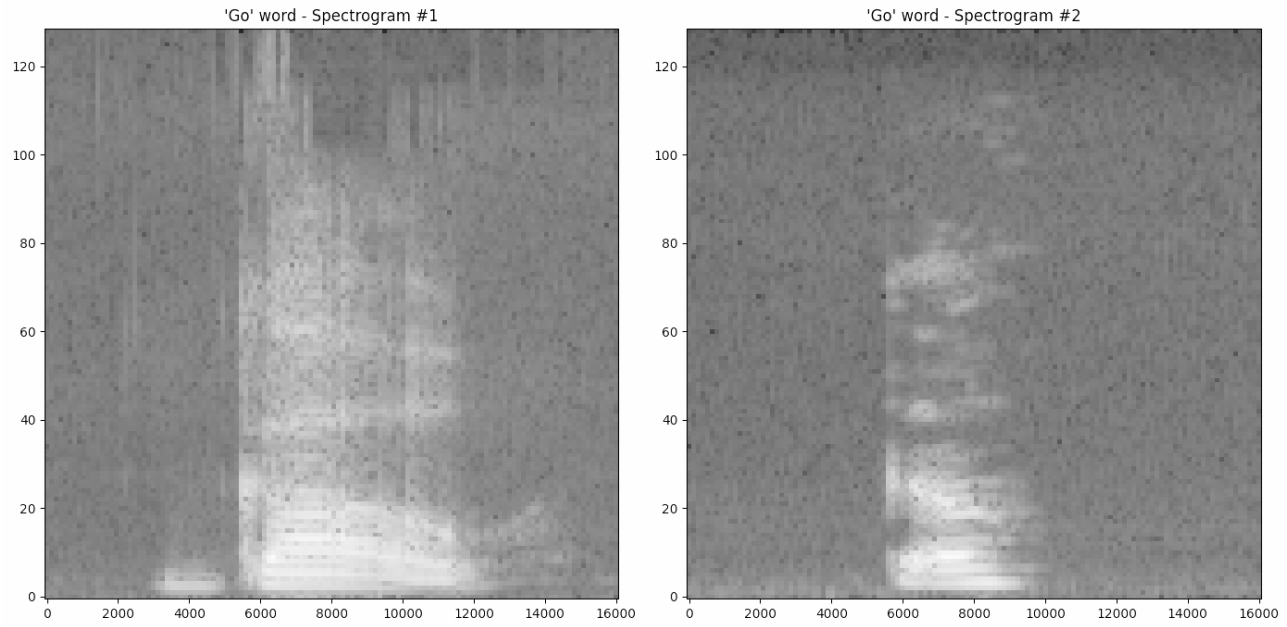


Рисунок 3 – спектрограмма, построенная для слова «Go»

Спектрограммы разных аудио-записей, построенные для одного слова обладают характерными общими чертами, которые можно классифицировать (рисунок 4).



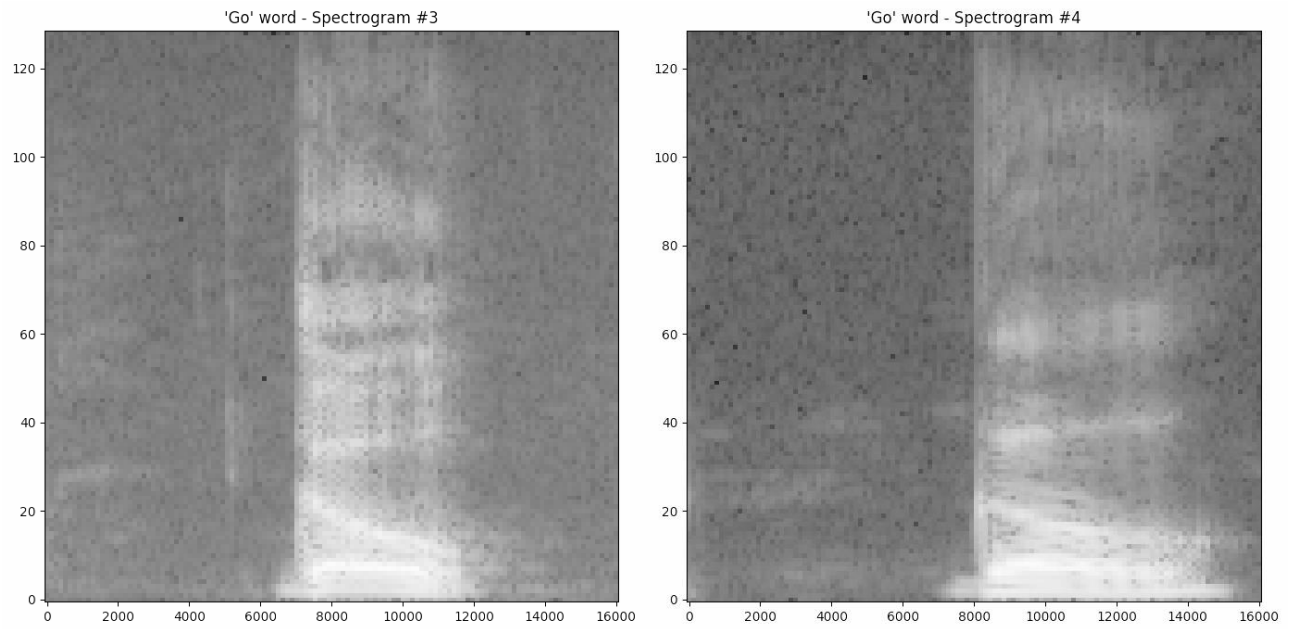


Рисунок 4 – Сравнение 4-х спектрограмм для слова «Go»

Таким образом, задача распознавания речи сводиться к задаче классификации изображений, что в свою очередь является типовой задачей, решение которой может быть достигнуто путем использования свёрточной нейронная сеть (Convolutional Neural Network) [4].

Свёрточной нейронная сеть – это тип архитектуры нейронной сети глубокого обучения, обычно используемой в компьютерном зрении [4].

Компьютерное зрение, в свою очередь, – это область искусственного интеллекта, которая позволяет компьютеру понимать и интерпретировать изображение или визуальные данные.

Рассмотрим пример архитектуры нейронной сети распознавания речи, построенной на базе свёрточной нейронная сеть [2], представленной на рисунке 5.

```
for spectrogram, _ in spectrogram_ds.take(1):
  input_shape = spectrogram.shape
print('Input shape:', input_shape)
num_labels = len(commands)

# Instantiate the `tf.keras.layers.Normalization` layer.
norm_layer = layers.Normalization()
# Fit the state of the layer to the spectrograms
# with `Normalization.adapt`.
norm_layer.adapt(data=spectrogram_ds.map(map_func=lambda spec, label: spec))

model = models.Sequential([
    layers.Input(shape=input_shape),
    # Downsample the input.
    layers.Resizing(32, 32),
    # Normalize.
    norm_layer,
    layers.Conv2D(32, 3, activation='relu'),
    layers.Conv2D(64, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.25),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_labels),
])

model.summary()
```

Рисунок 5 – Архитектура нейронной сети распознавания речи

В приведенном примере используется последовательная модель (Sequential model), обладающая следующими слоями [5]:

- Input – слой ввода данных;
- Resizing – слой изменение размера данных;
- norm_layer – слой нормализации данных;
- Conv2D – свёрточные слои, необходимые для анализа изображений;
- MaxPooling2D – слои извлечения важной информации, используются для уменьшения размера введенной информации;
- Dropout – слой, предотвращающий переобучение модели;
- Flatten – слой, конвертирующий данные в меньшую размерность;
- Dense – слой, производящий вывод на основе сформированной и проанализированной информации, используется для классификации.

Таким образом, описанные ранее действия по преобразованию аудио-файлов находят свое отражение в архитектуре слоев сверточной нейронной сети.

Заключение

В рамках этой статьи были рассмотрены основные принципы представления и преобразования аудио-файлов при работе с нейронными сетями распознавания речи. Рассмотрены наиболее распространенные форматы аудио-файлов, их отличительные черты и пригодность в использовании в датасетах. Рассмотрен процесс нормализации набора данных и его роль при обработке данных. Обосновано сведение задачи распознавания речи к задаче распознавания образов и классификации изображений. Информация затронутая в рамках данной статьи выступает в роли отправной точки при изучение работы нейронных сетей распознавания речи.

Библиографический список:

1. Speech recognition: The best audio files for transcription [Электронный ресурс] URL: https://filestar.com/blog/speech-recognition-the-best-audio-files-for-transcription (дата обращения: 24.09.2022).
2. Простое распознавание аудио: распознавание ключевых слов | Tensorflow Core [Электронный ресурс] URL: https://www.tensorflow.org/tutorials/audio/simple_audio?hl=ru (дата обращения: 25.09.2022).
3. Нормализация входных векторов [Электронный ресурс] URL: https://wiki.loginom.ru/articles/normalization.html (дата обращения: 25.09.2022).
4. Introduction to Convolution Neural Network [Электронный ресурс] URL: https://www.geeksforgeeks.org/introduction-convolution-neural-network/ (дата обращения: 26.09.2022).
5. Как работают слои в Keras, их типы и свойства [Электронный ресурс] URL:https://pythonru.com/biblioteki/sloi-keras-parametry-i-svojstva-keras-5?ysclid=lhxjru6o8k92446848 (дата обращения: 28.09.2022).