Санкт–Петербургский государственный университет

Кафедра компьютерного моделирования и многопроцессорных систем

Мирошниченко Александр Сергеевич

Выпускная квалификационная работа бакалавра

Разработка системы распознавания речевых команд при помощи методов машинного обучения

Направление 01.03.02

«Прикладная математика и информатика»

Научный руководитель,

кандидат физ.-мат. наук,

доцент

Козынченко В. А.

Санкт-Петербург

2021 г.

Содержание

Введение

В современном компьютеризированном мире огромное значение имеет взаимодействие человека с компьютером - ввод и вывод информации с устройства в понятной для человека форме. Один из способов внести информацию в компьютер - записать речь через микрофон, после чего можно обрабатывать данные в памяти, которые ее представляют. Обработка может быть совершенно разной, но особенно важно распознавать слова, которые произнес человек и давать им представление в виде текста. То есть давать такое представление речи, как если бы она была не произнесена голосом, а напечатана при помощи клавиатуры.

Решение такой задачи может быть использовано для различных целей. К примеру, можно переписываться с другим человеком по интернету текстовыми сообщениями, при этом вообще не прикасаясь к клавиатуре, управлять различными компьютерными интерфейсами при помощи голосовых команд и т.д.

Данная проблема была актуальна со времен появления компьютеров и остается таковой и по сей день. Особенно актуальна она стала в последнее время, когда появились качественные микрофоны, возрасла мощность вычислительных устройств, увеличилось количество информации, которой обмениваются люди через интернет.

Изначально, для решения данной задачи применялись такие алгоритмы, как скрытые Марковские модели, методы динамического программирования, методы дискриминантного анализа, основанные на Байесовской дискриминации и другие. Но с появлением нейронных сетей и многочисленных экспериментов с их использованием выяснилось, что задачу распознавания речи можно решать и при помощи нейросетевого подхода. И хоть и сами нейронные сети появились еще в прошлом веке, их популярность возросла только в последнее время в связи с ростом мощности компьютеров.

В данной работе предлагается рассмотреть решение задачи распознавания речи при помощи сверточной нейронной сети. В качестве решения подзадачи выделения характеристик речи предлагается алгоритм мел-частотных кепстральных коэффициентов, разработанный в 70-х годах прошлого века, учитывающий особенности слухового восприятия человеком.

Краткое содержание глав:

В главе 1 рассмотрены теоретические сведения о звуке, распознавании речи, речевых признаках.

В главе 2 рассмотрен алгоритм распознавания речи. Подробно описаны подзадачи предобработки звукового сигнала, выделения речевых признаков и самого блока распознавания.

В главе 3 приведены результаты вычислений - результаты обучения и тестирования нейронной сети, а также инструменты, которые были использованы для реализации алгоритма.

Постановка задачи

Пусть $X = \{x_1, ..., x_p, ...\}$ - множество объектов речи. Оно состоит из векторов амплитуд $x_i = \{x_i^1, ..., x_i^{k_i}\}$, $i = \overline{1, \inf}$. Здесь k_i - количество записанных амплитут в i-м объекте. У каждого объекта речи есть своя частота дискретизации w_i. Само по себе множество объетов бесконечно, однако известны значения первых p элементов. Обозначим их через $\widehat{X} = \{x_1, ..., x_p\}$.

Пусть $Y = \{y_1, ..., y_p, ...\}$ - множество скалярных меток, а $\widehat{Y} = \{y_1, ..., y_p\}$ - множество известных скалярных меток для первых p объектов речи. Таким образом известно отображение $\widehat{X} \to \widehat{Y}$, описываемое парами значений $\widehat{Z} = \{(x_1, y_1), ..., (x_p, y_p)\}$.

Необходимо разработать алгоритм, который бы строил отображение $Z = X \to Y$ на всем множестве X.

Для того, чтобы решить поставленную задачу, необходимо разбить ее на следующие подзадачи и последовательно решить их:

- Провести предобработку первоначальных данных, содержащих звуковой сигнал в виде наборов амплитуд
- Разработать алгоритм распознавания объектов речи
- Реализовать алгоритм распознавания объектов речи
- Провести вычислительные эксперименты и выяснить, какой метод решения задачи является наиболее эффективным

Выбор инструментов для решения задачи

Для решения задачи был выбран язык программирования Python 3.8. Предобработку данных было решено реализовывать при помощи Python 3.8. В качестве библиотеки для реализации нейронной сети была выбрана библиотека Keras, включенная в библиотеку Tensorflow 2.4.1.

Для создания датасета был разработан веб-сервис на NodeJS, Javascript, HTML, CSS, который позволяет записывать команды и сохранять их в нужном для программы предобработки формате. Также это позволило записать необходимое количество дикторов, которые смогли довольно быстро наговорить команды.

Обзор литературы

Глава 1. Теоретические сведения

Глава 2. Описание решения

Глава 3. Результаты вычислений

Было проведено 3 вычислительных эксперимента для нейронной сети типа CNN. Структура сети приведена на рисунке 1.

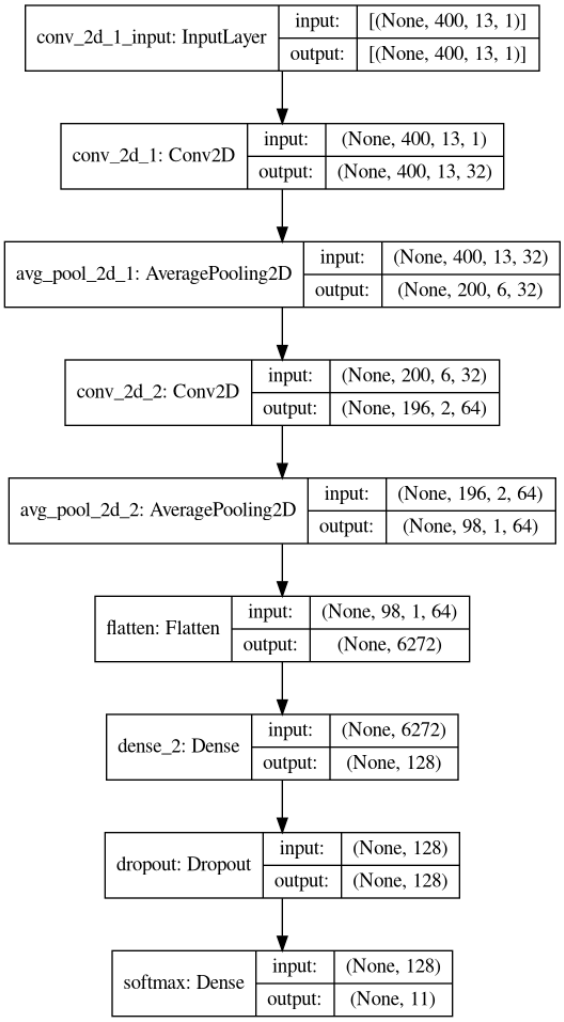

Рис. 1: Структура модели нейронной сети типа CNN

Все звуковые файлы были предобработаны при помощи алгоритма MFCC. Звуковая дорожка делится на фреймы. Каждый фрейм - отрезок звуковой дорожки длительностью 20 мс. Каждый фрейм начинается с момента (10 мс. $\times$ номер_фрейма), нумерация начинается с 0. Для каждой звуковой дорожки количество фреймов - 400. Если количество фреймов у дорожки меньше 400, то слева и справа добавляются нули. Это число было выбрано как максимально возможное количество фреймов для всех дорожек. Количество коэффициентов в алгоритме MFCC - 13, количество фильтров - 26. В итоге размерность данных, поступающих на вход нейронной сети - 400×13.

Датасет состоит из 6 дикторов. Каждый диктор записал 11 команд : 'back', 'down', 'menu', 'off',

'on', 'open', 'play', 'power', 'stop', 'up', 'volume'.

Диктор	Тип голоса	Кол-во звук. дорожек на каждую команду	Сумм. кол-во звук. дорожек
speaker1	Мужской	50	550
speaker2	Мужской	40	440
speaker3	Мужской	40	440
speaker4	Мужской	40	440
speaker5	Мужской	50	550
speaker6	Женский	50	550

Первый эксперимент: нейронная сеть обучается на первом дикторе с мужским голосом, тестирование производится на каждом дикторе.

Второй эксперимент: нейронная сеть обучается на всех дикторах с мужским голосом, тестирование производится на каждом дикторе.

Третий эксперимент: нейронная сеть обучается на всех дикторах, тестирование производится на каждом дикторе.

Датасет предварительно разделяется на тренировочную и тестовую части. На тренировочную часть отводится 70% данных диктора, на тестовую часть - 30%. В процессе тренировки после каждой эпохи тренировочные данные перемешиваются. 15% тренировочных данных в каждой эпохе - валидационные. В качестве метрики для оценки эффективности была выбрана метрика точности (accuracy), а для валидации - функция потерь категориальной кросс-энтропии (val_loss). Алгоритм оптимизации - Adam. Максимальное количество эпох - 50. Если значение метрики val_loss не уменьшается в течение 20 эпох, то обучение останавливается.

Графики обучения для каждого из экспериментов приведены на рисунках 2, 3, 4.

В конце каждого эксперимента проводится тестирование нейронной сети. А в случае обучения на all_speakers помимо тестирования производится построение матрицы ошибок (confusion matrix) для каждого диктора и для каждого из четырех пороговых значений: 0.5, 0.6, 0.7, 0.8. Матрицы представлены на рисунке 5.

Результаты тестирования представлены в таблице 1. Обозначения, которые используются:

all_speakers = [speaker1, speaker2, speaker3, speaker4, speaker5, speaker6]

all_male_speakers = [speaker1, speaker2, speaker3, speaker4, speaker5]

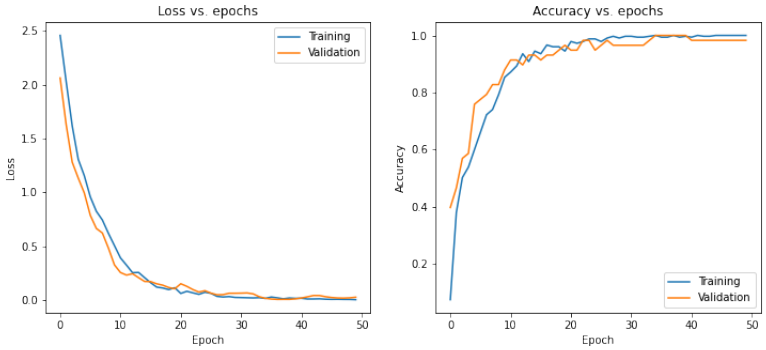

Рис. 2: Графики функции потерь и точности в течение обучения на speaker1

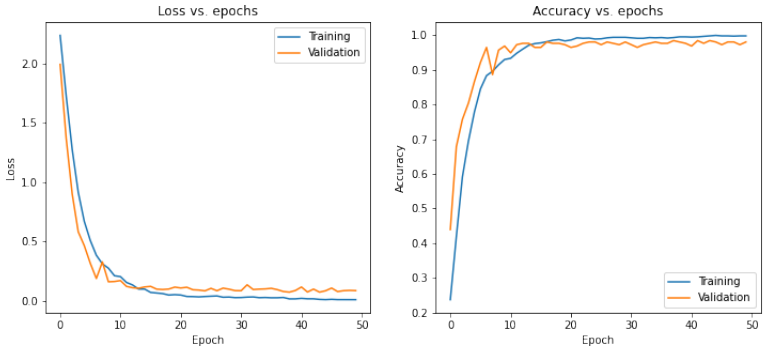

Рис. 3: Графики функции потерь и точности в течение обучения на all_male_speakers

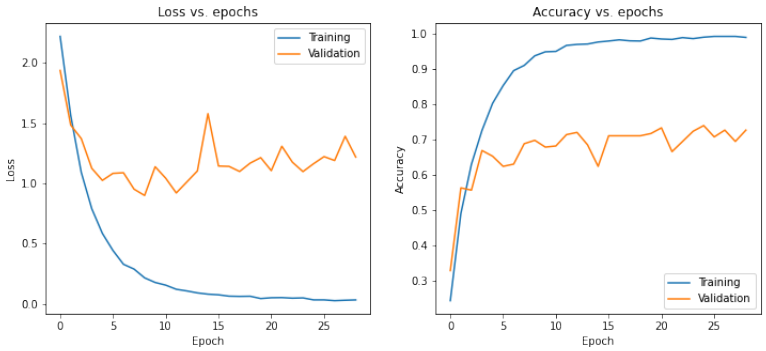

Рис. 4: Графики функции потерь и точности в течение обучения на all_speakers

train_data	test_speaker	cnn_loss	mlp_loss	cnn_accuracy	mlp_accuracy
speaker1	speaker1	0.086	0.063	0.994	0.982
speaker1	speaker2	12.41	8.032	0.114	0.121
speaker1	speaker3	4.69	3.313	0.394	0.288
speaker1	speaker4	8.181	7.442	0.212	0.258
speaker1	speaker5	3.106	2.998	0.648	0.612
speaker1	speaker6	18.053	9.348	0.182	0.139
all_male_speakers	speaker1	0.034	0.031	0.988	0.994
all_male_speakers	speaker2	0.113	0.236	0.962	0.917
all_male_speakers	speaker3	0.01	0.071	1.0	0.985
all_male_speakers	speaker4	0.44	0.743	0.924	0.871
all_male_speakers	speaker5	0.054	0.124	0.982	0.982
all_male_speakers	speaker6	9.527	15.87	0.242	0.255
all_speakers	speaker1	0.056	0.043	0.988	0.988
all_speakers	speaker2	0.131	0.199	0.97	0.932
all_speakers	speaker3	0.038	0.033	0.985	0.992
all_speakers	speaker4	0.35	0.605	0.939	0.879
all_speakers	speaker5	0.019	0.051	0.994	0.982
all_speakers	speaker6	1.361	2.54	0.709	0.63

Таблица 1: Результаты вычислений

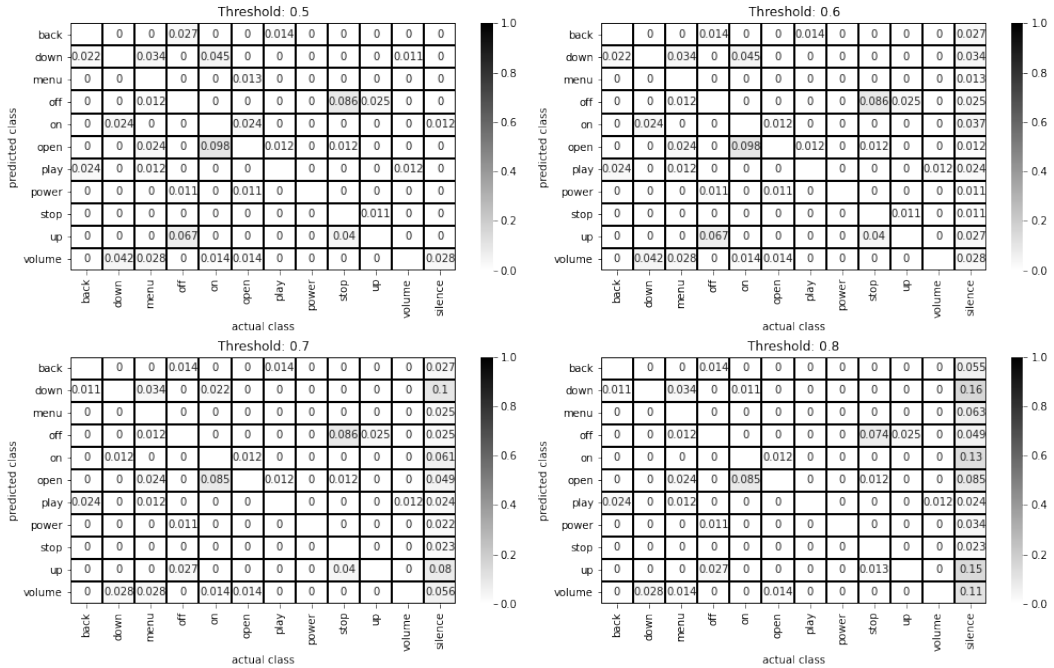

Рис. 5: Матрицы ошибок для случая обучения на all_speakers

Видно, что при обучении только на одном дикторе, распознавание на всех остальных работает плохо.

При обучении только на мужских голосах, распознавание на женском работает лучше, чем при обучении на одном, но все-равно очень плохо.

При обучении на всех голосах, распознавание на каждом голосе дает приемлемую точность. Однако стоит отметить, что если большая часть голосов в тренировочной части - мужские, то на женском голосе распознавание будет работать хуже, чем на мужских.

Выводы

Заключение

В данной работе:

- Проведена предобработка звуковых дорожек, содержащих команды в wav файлах
- Разработан алгоритм распознавания речевых команд
- Реализован алгоритм распознавания речевых команд
- Проведены вычислительные эксперименты, в результате которых показана работоспособность и эффективность работы алгоритма распознавания речевых команд.

Список использованных источников

[1] Aurélien G. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems / Aurélien G. — 2nd Edition — O'Reilly Media, 2019.

[2] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-yiin Chang, Tara Sainath Deep Learning for Audio Signal Processing // Journal of Selected Topics of Signal Processing, Vol. 13, No. 2, May 2019, pages 206–219.

[3] Документация TensorFlow [Электронный ресурс]. — Режим доступа: https://www.tensorflow.org/api_docs/python/tf

[4] Портал ML Glossary [Электронный ресурс]. — Режим доступа: https://ml-cheatsheet.readthedocs.io

[5] Курс на платформе Coursera [Электронный ресурс]. — Режим доступа: https://www.coursera.org/learn/getting-started-with-tensor-flow2

[6] Страница на Википедии [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Fourier_transform

[7] Страница на Википедии [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Mel_scale

Приложение

Ссылка на репозиторий с программой веб-сервисом для записи датасета, состоящего из звуковых файлов: `https://gitlab.com/polotent/commandrecorder`

Ссылка на репозиторий с программой предобработки данных, обучением и тестированием нейронной сети: `https://gitlab.com/polotent/boxy`