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. Одним из способов внести информацию в комьютер - записать речь, произнося ее в микрофон, после чего ее можно обрабатывать. Обработка может быть совершенной разной, но особенно важно распознавать слова, которые произнес человек и преобразовывать и дать им представление в виде текста. 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 с 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, появились персональные компьютеры с достаточной производительностью для этой задачи.

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 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 десятилетее, в связи с ростом мощности компьютеров.

В данной работе рассмотрено решение задачи распознавания речи при помощи сверточной нейронной сети. В качестве решения подзадачи выделения характеристик речи был выбран алгоритм мелчастотных кепстральных коэффициентов, разработанный в 70-х годах прошлого века. Он учитывает особенности слухового восприятия человеком.

Постановка задачи

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.

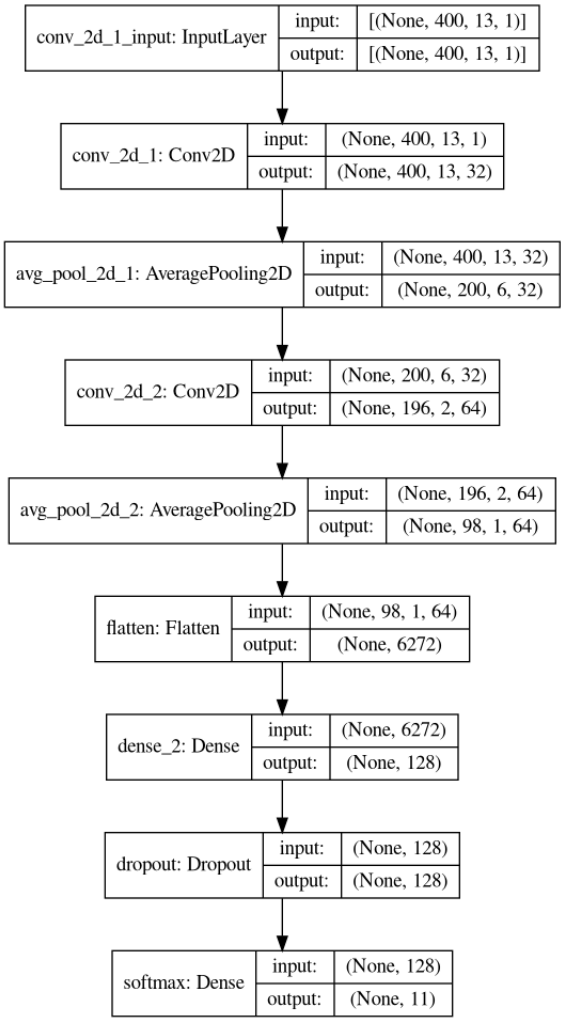

Рис. 1: Структура модели нейронной сети типа CNN

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]

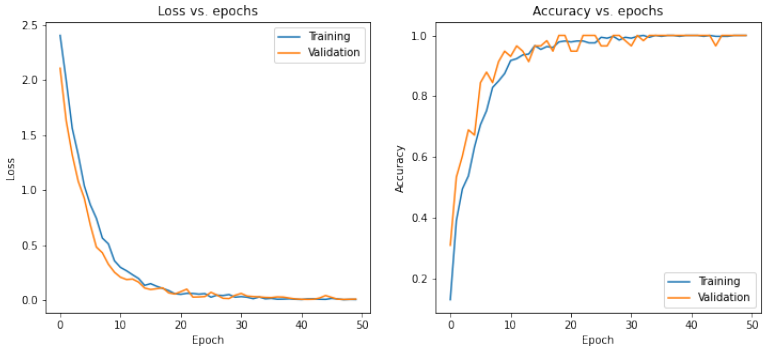

Рис. 2: Графики функции потерь и точности в течение обучения на speaker1

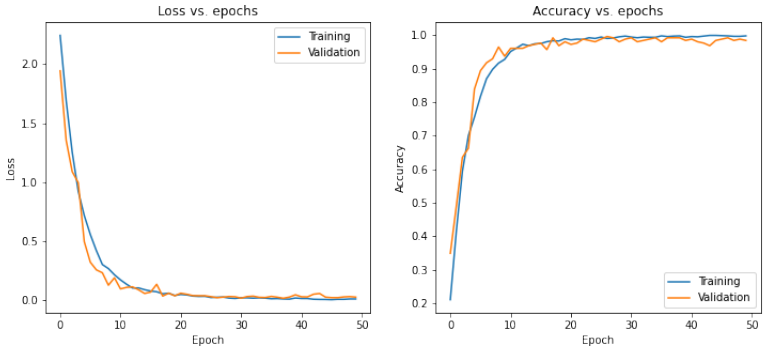

Рис. 3: Графики функции потерь и точности в течение обучения на all_male_speakers

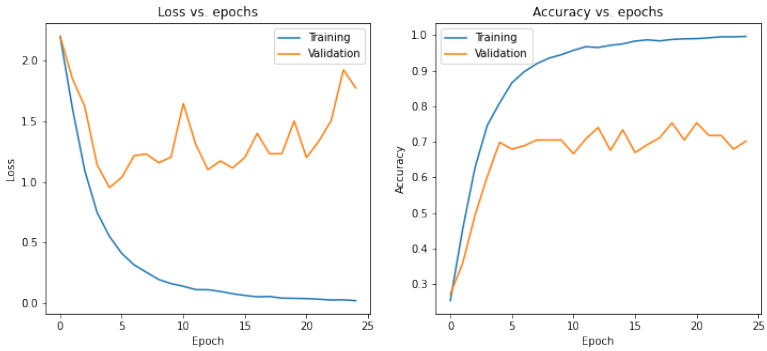

Рис. 4: Графики функции потерь и точности в течение обучения на all_speakers

train_data	test_speaker	cnn_loss	cnn_accuracy
speaker1	speaker1	0.056	0.994
speaker1	speaker2	10.015	0.167
speaker1	speaker3	4.832	0.341
speaker1	speaker4	8.05	0.197
speaker1	speaker5	2.526	0.618
speaker1	speaker6	23.892	0.145
all_male_speakers	speaker1	0.062	0.994
all_male_speakers	speaker2	0.252	0.962
all_male_speakers	speaker3	0.092	0.977
all_male_speakers	speaker4	0.846	0.909
all_male_speakers	speaker5	0.028	0.982
all_male_speakers	speaker6	10.637	0.333
all_speakers	speaker1	0.058	0.988
all_speakers	speaker2	0.153	0.947
all_speakers	speaker3	0.085	0.985
all_speakers	speaker4	0.561	0.909
all_speakers	speaker5	0.015	0.988
all_speakers	speaker6	2.144	0.679

Таблица 1: Результаты вычислений

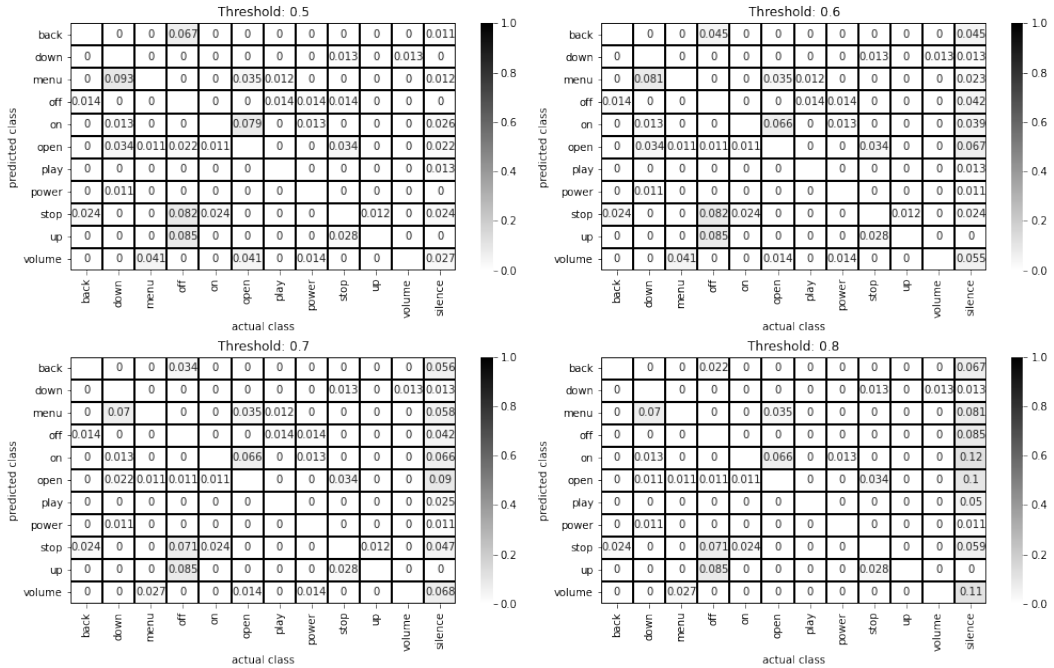

Рис. 5: Матрицы ошибок для случая обучения на all_speakers

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.

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`