

Задание 1. Программирование

Описание задачи

Найти непрерывный подмассив в массиве, содержащий хотя бы одно число, который имеет наибольшую сумму.

Условия

Необходимо написать программу с функцией `findMaxSubArray(A)`, принимающей на вход массив целых чисел `A` ненулевой длины и возвращающей непрерывный подмассив массива `A` ненулевой длины, который имеет наибольшую сумму среди всех непрерывных подмассивов массива `A`.

Язык программирования: C++ или python

Использование дополнительных библиотек и функций: не разрешается

Пример

На вход подается массив `[-2, 1, -3, 4, -1, 2, 1, -5, 4]`

На выходе функции ожидается массив `[4, -1, 2, 1]`, имеющий максимальную сумму среди всех подмассивов равную 6.

Задание 2. ML

В рамках данной задачи предлагается решить проблему, возникающую при работе с зашумленным звуком. Решение предполагает построение модели, использующее стандартные ML практики или нейронные сети, ее обучение и тестирование.

Описание задачи

Необходимо реализовать алгоритм, позволяющий определить, является ли аудиозапись зашумленной или нет.

Описание данных

Для обучения и тестирования модели будет предоставлена выборка mel-спектрограмм, построенным по чистым и зашумленным звуковым файлам с голосом человека. Чистым звуком считается звук голоса без посторонних шумов (возможно с паузами). Шумом на зашумленных аудиозаписях может являться любой посторонний звук, который можно услышать в повседневной жизни - звонок телефона, проезжающая машина, чайник, смех и т.д.

В качестве данных будет предоставлено 14 тысяч пар mel-спектрограмм (чистый звук + зашумленный звук), соответствующих звуковым файлам, все файлы разных длительностей. Из них 12 тысячи - тренировочная выборка, 2 - валидационная.

Тестирование будет проводиться на закрытой выборке, состоящей из 2 тысяч спектрограмм.

Данные можно скачать по ссылкам: [тренировочная выборка](#), [валидационная выборка](#).

Пример данных доступен в jupyter notebook по [ссылке](#).

Все данные разделены на чистые и зашумленные. Внутри каждой категории есть разделение по спикерам. Для каждого спикера есть набор из чистых данных и набор из зашумленных данных. Название файла с зашумленными данными соответствует названию файла с оригинальными данными.

Топология данных для тренировочной выборки (для тестовой идентична):

```
- clean\
- - speaker1_id\
- - - spec_sample_1_1.npy
- - - spec_sample_1_2.npy
- - - ...
- - speaker2_id\
- - - spec_sample_2_1.npy
- - - spec_sample_2_2.npy
- - - ...
- - ...
- noise\
- - speaker1_id\
- - - spec_sample_1_1.npy
- - - spec_sample_1_2.npy
- - - ...
- - speaker2_id\
- - - spec_sample_2_1.npy
- - - spec_sample_2_2.npy
- - - ...
- - ...
```

Метрики

Метрика для алгоритма классификации: accuracy

Дополнительные условия

Язык программирования: любой

Библиотеки\фреймворки для ML: любые

Результатом задачи должны являться:

- 1) Скрипт(ы) для обучения
- 2) Описание алгоритма
- 3) Скрипт для тестирования, принимающий на вход файл MEL спектрограммы и выдающий ответ, и описание возможных взаимодействий с этим скриптом.