**Обнаружение слова пробуждения**

Начнем с обнаружения слова пробуждения. Нам нужно создать «пробуждающее» слово, услышав которое, система будет знать, что пора исполнять какое-то действие. Идеальный вариант использования на встроенных устройствах – обратиться к TensorFlow и TensorFlow Lite.

**Данные для обучения**

Наша первая цель - найти данные для обучения модели. Мы можем использовать набор данных речевых команд.

<https://www.tensorflow.org/datasets/catalog/speech_commands>

Этот набор данных содержит более 100 000 аудиофайлов, состоящих из набора из 20 основных командных слов, таких как «Вверх», «Вниз», «Да», «Нет» и набора дополнительных слов. Каждый из отсчетов длится 1 секунду.

Одно из этих слов, в частности, выглядит как хороший кандидат на слово для пробуждения - я решил использовать слово «Марвин» в своем слове для пробуждения как дань уважения андроиду из Путеводителя автостопщиков по Галактике.

Вот несколько примеров слова «Марвин»:

Марвин1 | Марвин1

А вот еще несколько случайных слов из набора данных:

Вперед | Влево | Вправо

Чтобы расширить набор данных, вы также можете записать фоновый шум окружающей среды, я записал несколько часов домашнего шума и телешоу, чтобы предоставить большое количество случайных данных.

**Функции**

Имея данные для обучения, нам нужно подумать, с какими функциями мы собираемся обучать нашу нейронную сеть. Маловероятно, что передача необработанной звуковой волны в нашу нейронную сеть даст нам хорошие результаты.

Популярным подходом к распознаванию слов является перевод проблемы в проблему распознавания изображений.

Нам нужно превратить наши аудиосэмплы во что-то похожее на изображение - для этого мы можем взять спектрограмму.

Чтобы получить спектрограмму аудиосэмпла, мы разбиваем его на небольшие части, а затем выполняем дискретное преобразование Фурье для каждого раздела. Это даст нам частоты, которые присутствуют в этом фрагменте звука.

Объединение этих частотных срезов вместе дает нам спектрограмму образца.

В папке модели вы найдете несколько записных книжек Jupyter. Следуйте инструкциям по установке на README.md, чтобы настроить локальную среду.

Блокнот Generate Training Data.ipynb содержит код, необходимый для извлечения наших функций из наших аудиоданных.

Следующая функция может использоваться для генерации спектрограммы из аудиосэмпла:

def get\_spectrogram(audio):

# normalise the audio

audio = audio - np.mean(audio)

audio = audio / np.max(np.abs(audio))

# create the spectrogram

spectrogram = audio\_ops.audio\_spectrogram(audio,

window\_size=320,

stride=160,

magnitude\_squared=True).numpy()

# reduce the number of frequency bins in our spectrogram to a more sensible level

spectrogram = tf.nn.pool(

input=tf.expand\_dims(spectrogram, -1),

window\_shape=[1, 6],

strides=[1, 6],

pooling\_type='AVG',

padding='SAME')

spectrogram = tf.squeeze(spectrogram, axis=0)

spectrogram = np.log10(spectrogram + 1e-6)

return spectrogram