Необходимые библиотки/фреймворки

Pytorch

Scipy

Numpy

Matplotlib

Пример запуска:

1. Подготовка данных **python prepare\_data.py ./pulses\_set ./features\_set**
2. Обучение сети **python bp\_detection.py ./features/ ./models/ model\_11 10**
3. **ОБРАБОТКА ПОСЛЕДОВАТЕЛЬНОСТЕЙ. prepare\_data.py**

Запустить:

python prepare\_data.py <path\_to\_pulses> <path\_to\_output\_features>

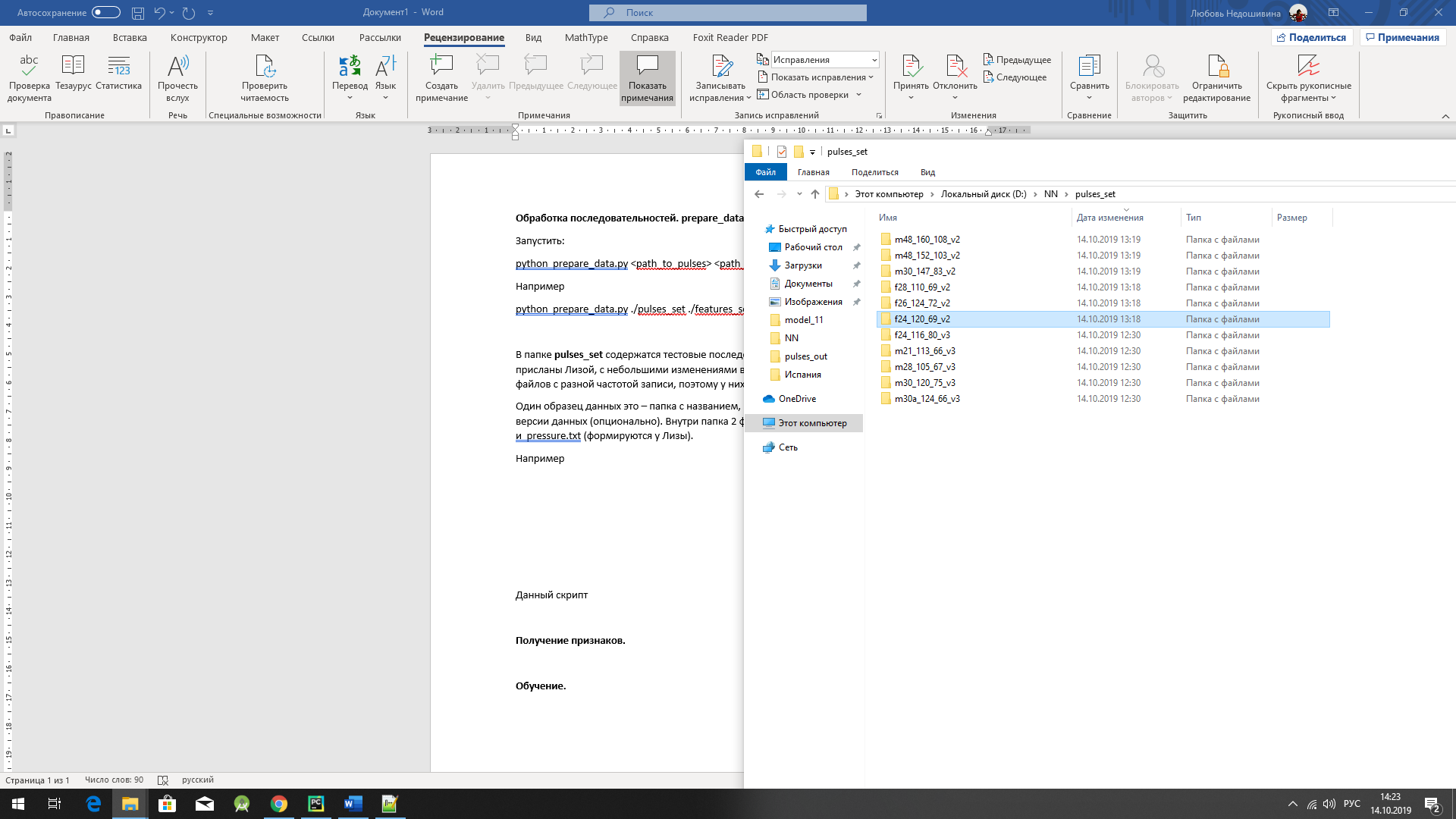
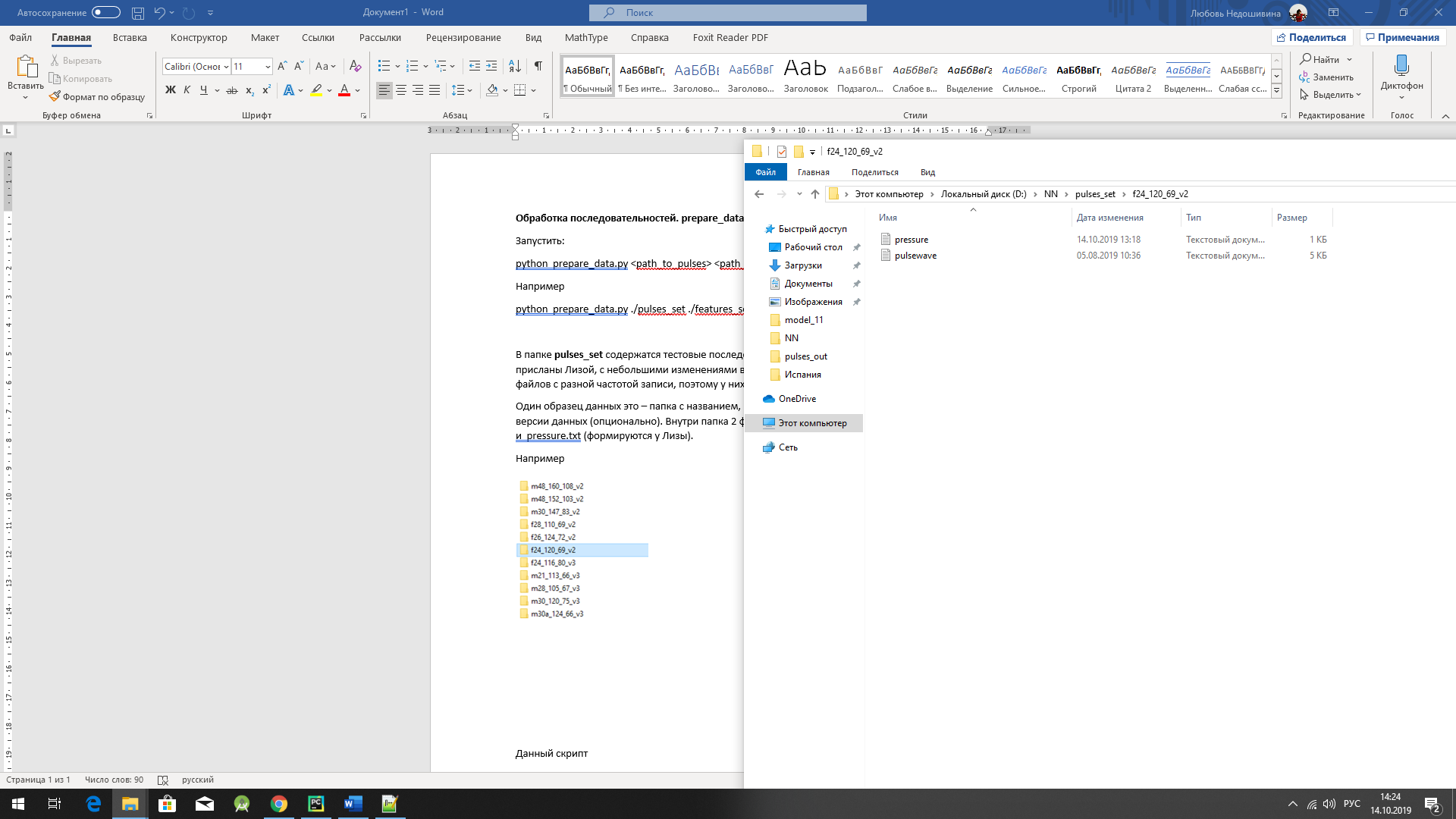
Например

python prepare\_data.py ./pulses\_set ./features\_set

В папке **pulses\_set** содержатся тестовые последовательности в том же формате, в каком были присланы Лизой, с небольшими изменениями в названиях (так как были посланы две группы файлов с разной частотой записи, поэтому у них есть окончание с номером версии \_v2 и \_v3).

Один образец данных это – папка с названием, содержащим пол, возраст, давление и номер версии данных (опционально). Внутри папка 2 файла с фиксированными названиями pulsewave.txt и pressure.txt (формируются у Лизы).

Например

F24\_120\_69\_v2 можно расшифровать как:

F – female

24 – возраст

120 – верхнее давление

69 – нижнее давление

\_v2 – номер версии данных

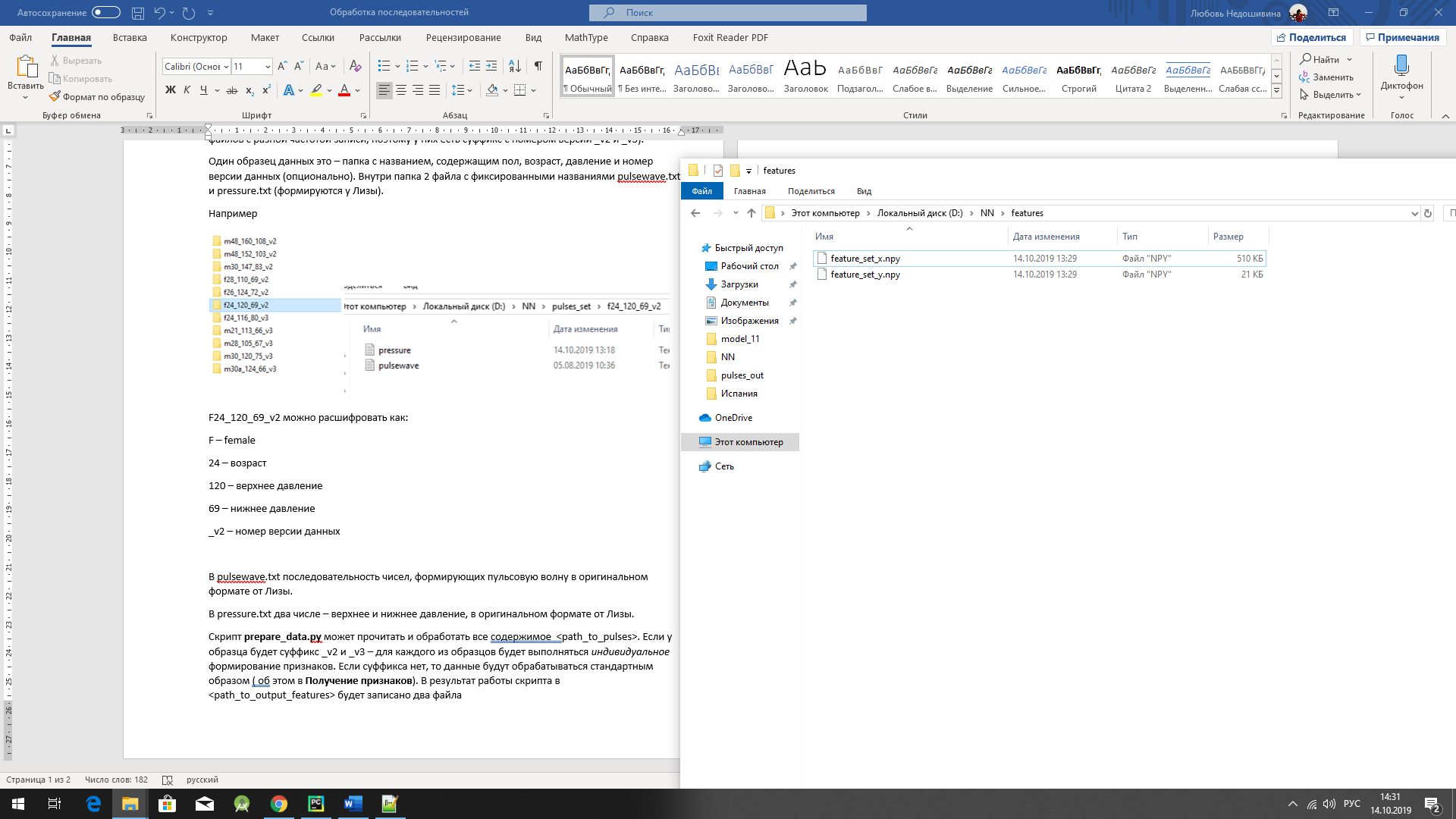
В pulsewave.txt последовательность чисел, формирующих пульсовую волну в оригинальном формате от Лизы.

В pressure.txt два числе – верхнее и нижнее давление, в оригинальном формате от Лизы.

Скрипт **prepare\_data.py** может прочитать и обработать все содержимое <path\_to\_pulses>. Если у образца будет окончание \_v2 и \_v3 – для каждого из образцов будет выполняться *индивидуальное* формирование признаков. Если суффикса нет, то данные будут обрабатываться стандартным образом (об этом в **Получение признаков**).

*Таким образом, чтобы добавить новые данные, нужно рядом с текущим набором данных, создать папку с аналогичным названием, положить в нее файлы pulsewave.txt и pressure.txt.*

В результате работы скрипта в <path\_to\_output\_features> будет записано два файла в одном из форматов Python для хранения данных \*.npy.



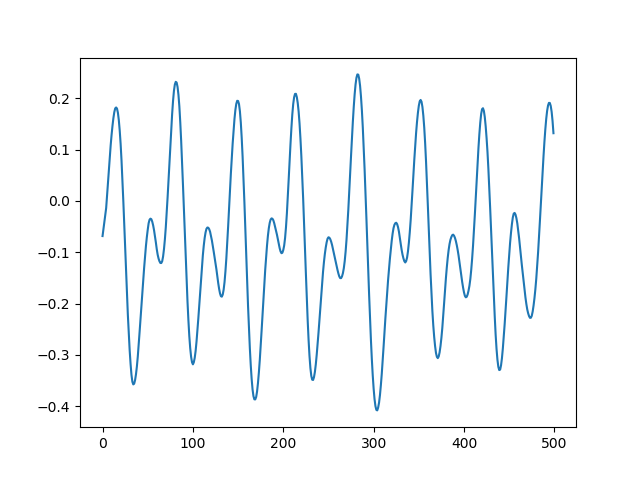
**feature\_set\_x.npy** содержит набор полученных из пульсовых волн векторов признаков.

**feature\_set\_y.npy** содержит соответствующие значения давления.

1. **ПОЛУЧЕНИЕ ПРИЗНАКОВ в prepare\_data.py**

**Стандартная обработка пульсовой волны**

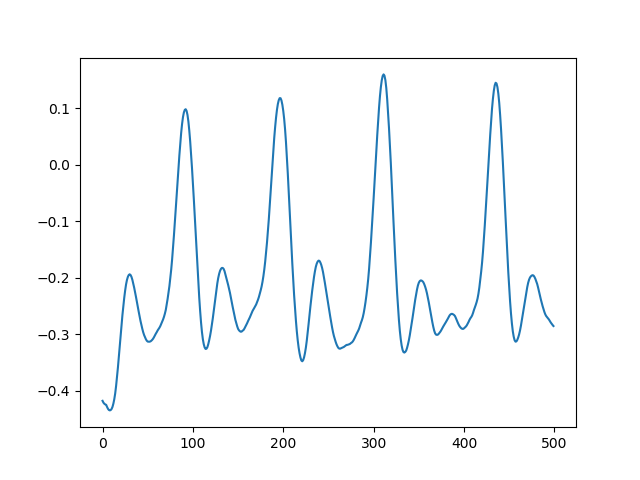
Ожидаемая длина пульса 60 отсчетов. В данном случае волна последовательно нарезается на кусочки в 60 отсчетов, начиная с 0-го отсчета, как показано на рисунке. Далее для каждого пульса считается амплитудный спектр. Вектор признаков – это 50 первых частот амплитудного спектра.



**60**

**120**

**Индивидуальная обработка пульсовой волны (пример)**



**151**

**51**

В этом случае выводится график пульсовой волны, выбирается начало отчета (в данном примере это 51), далее по волне находится начало следующего пульса – в данном случае это примерно 151, таким образом длина пульса составляет ~100 отсчетов. Поэтому этот файл обрабатывается так:

Начиная с 51 отсчета, последовательно берутся кусочки длиной 100 отчетов. Далее выполняется понижение размерности кусочка со 100 до 60 отсчетов с помощью функции библиотеки scipy.signal resample. Для полученного уменьшенного пульса считается амплитудный спектр. Вектор признаков – это 50 первых частот амплитудного спектра.

1. **ОБУЧЕНИЕ.**

Скрипт bp\_detection.py

python bp\_detection.py <path\_to\_features> <path\_to\_models> <model\_name> <number\_of\_training\_epochs>

Например

**python bp\_detection.py ./features/ ./models/ model\_11 10**

./features/ - Путь до директории с файлами features\_x.npy и features\_y.npy

./models/ - путь до директории, куда будет записываться модели на каждой 10й эпохе обучения, финальная модель и csv файл с результатами валидации на последней эпохе обучения.

model\_11 – название модели

10 – количество эпох обучения

В данном скрипте загружаются подготовленный набор признаков и формируются обучающая и валидационная выборка.

train\_x, train\_y, valid\_x, valid\_y = get\_train\_and\_valid\_set(args[1],if\_random=**False**, train\_share=0.8)

Доля обучающей выборки от общего набора признаков train\_share=0.8 – 80% от всех признаков составляют обуч. выборку.

if\_random=False – означает, что разбиение будет производиться без перемешивания признаков. Если True. То набор данных сначала случайным образом перемешается, а затем разобьется на 2 выборки.