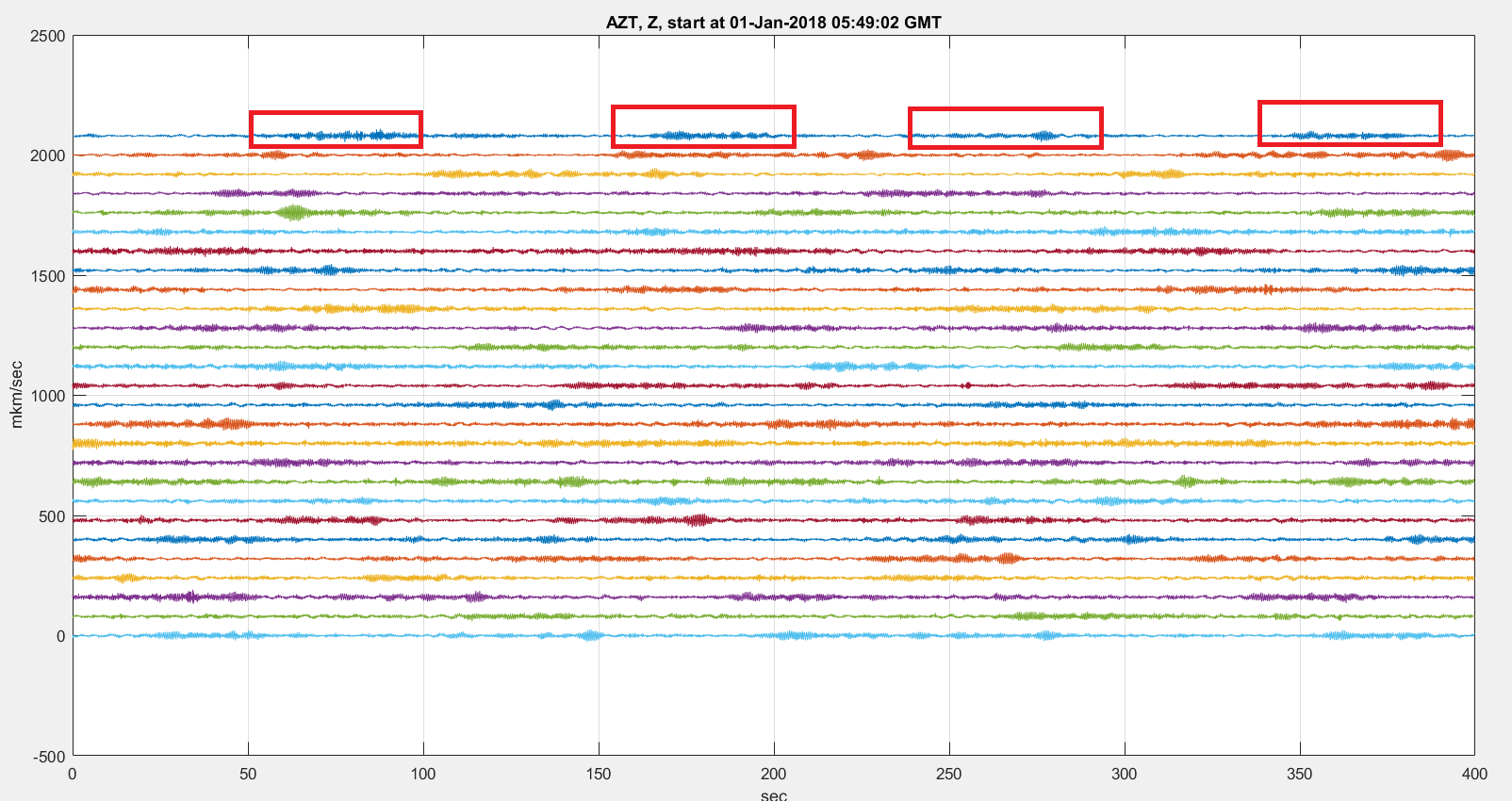
Задача: распознавание сигнла от прохождения поезда на сейсмограмме.

Сейсмограмма:

Красными квадратами отмечены отрезки времени, во время которых фиксируется сигнал от поезда. Надо отметить, что в данном случае сигнал содержит мало шума и сигнал от поезда виден глазом, но такие случаи бывают редко.

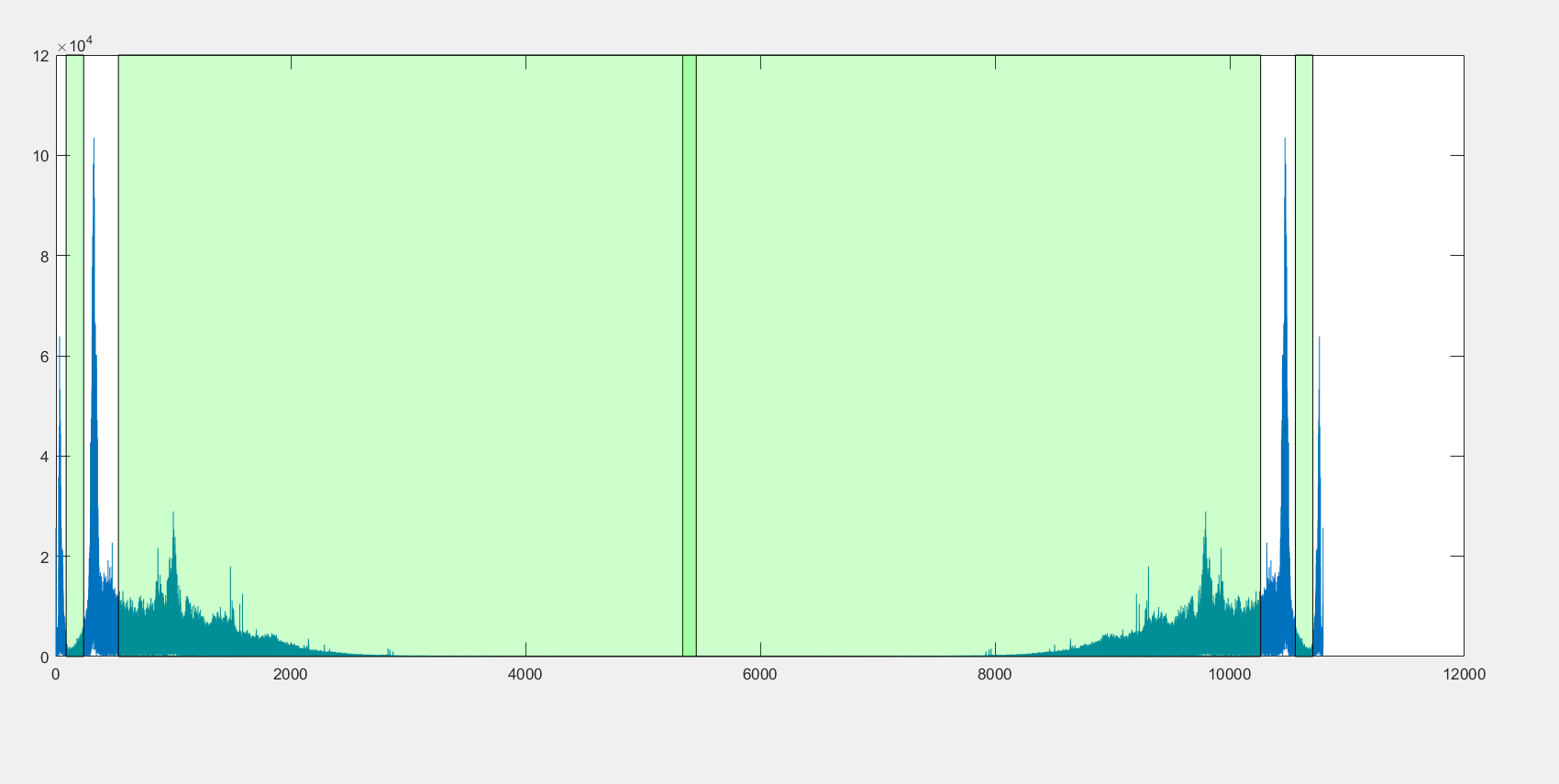
Если посмотеть на фурье-образы участков с сигналом от поезда и без него, можно заметить, что на некоторых частотах они сильно отличаются.

|  |  |
| --- | --- |
|  |  |

На рисунках показана разница фурье-образа для участка (зеленое окно) в зависимости попадает туда сигнал от поезда или нет(сигнал возведен в квадрат, чтобы глазом лучше различались участки с поездом по амплитуде).

Соответственно можно предположить, что можно создать некий фильтр, применив который на сигнал, можно будет обнаружить участки, содержащие сигал от поезда.

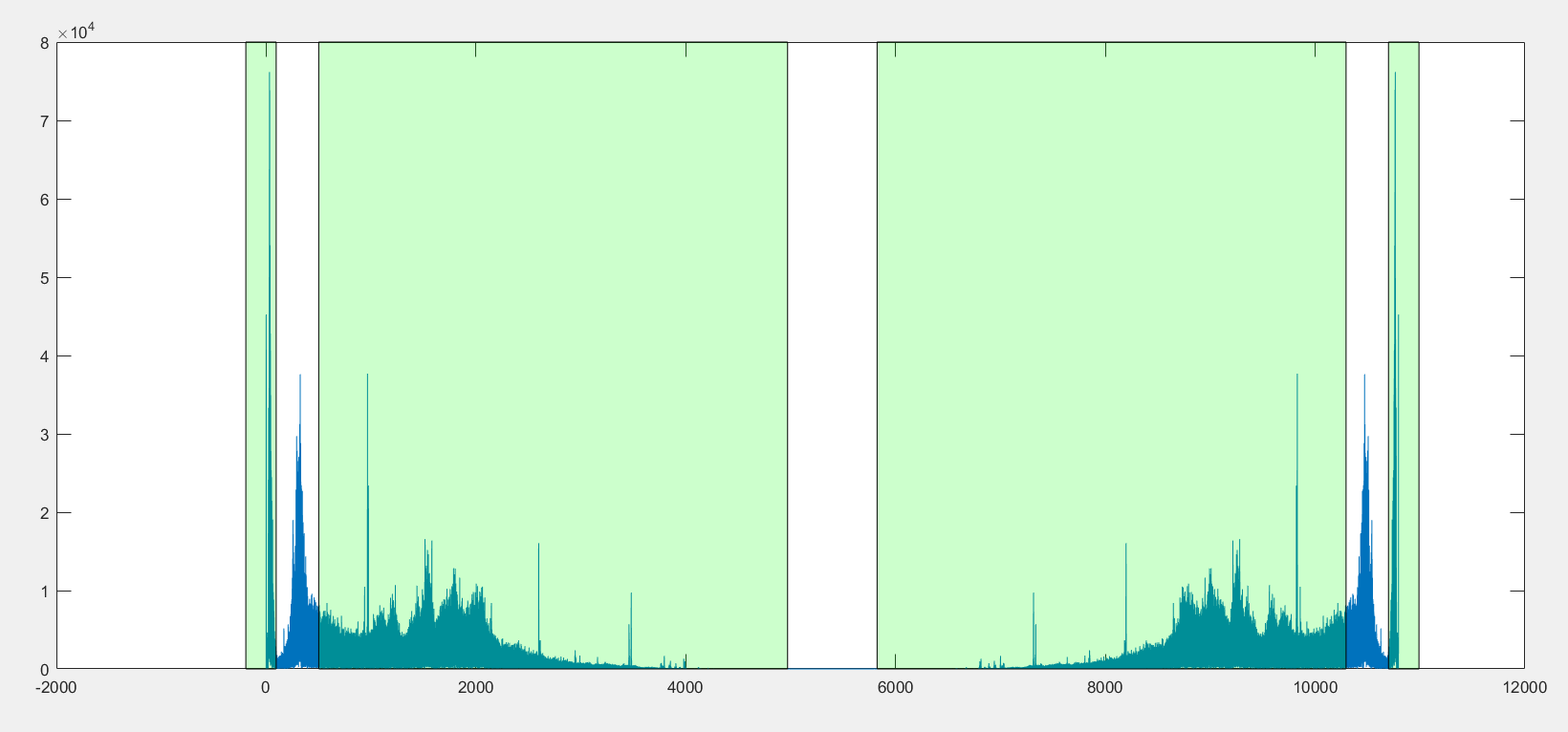
Например, если в фурье-образе сигнала, показанном на следующем рисунке занулить выделенные зеленым участки,



то картина изменится следующим образом(показаны фрагменты):

|  |  |
| --- | --- |
| До: | После: |

На более зашумленном сигнале результат заметнее



|  |  |
| --- | --- |
|  |  |

Задача состоит в том, чтобы найти некий оптимальный фильтр, применив который можно наилучшим образом выделить сигнал. Для этого предполагается использовать библиотеку для обучения нейросетей TensorFlow.

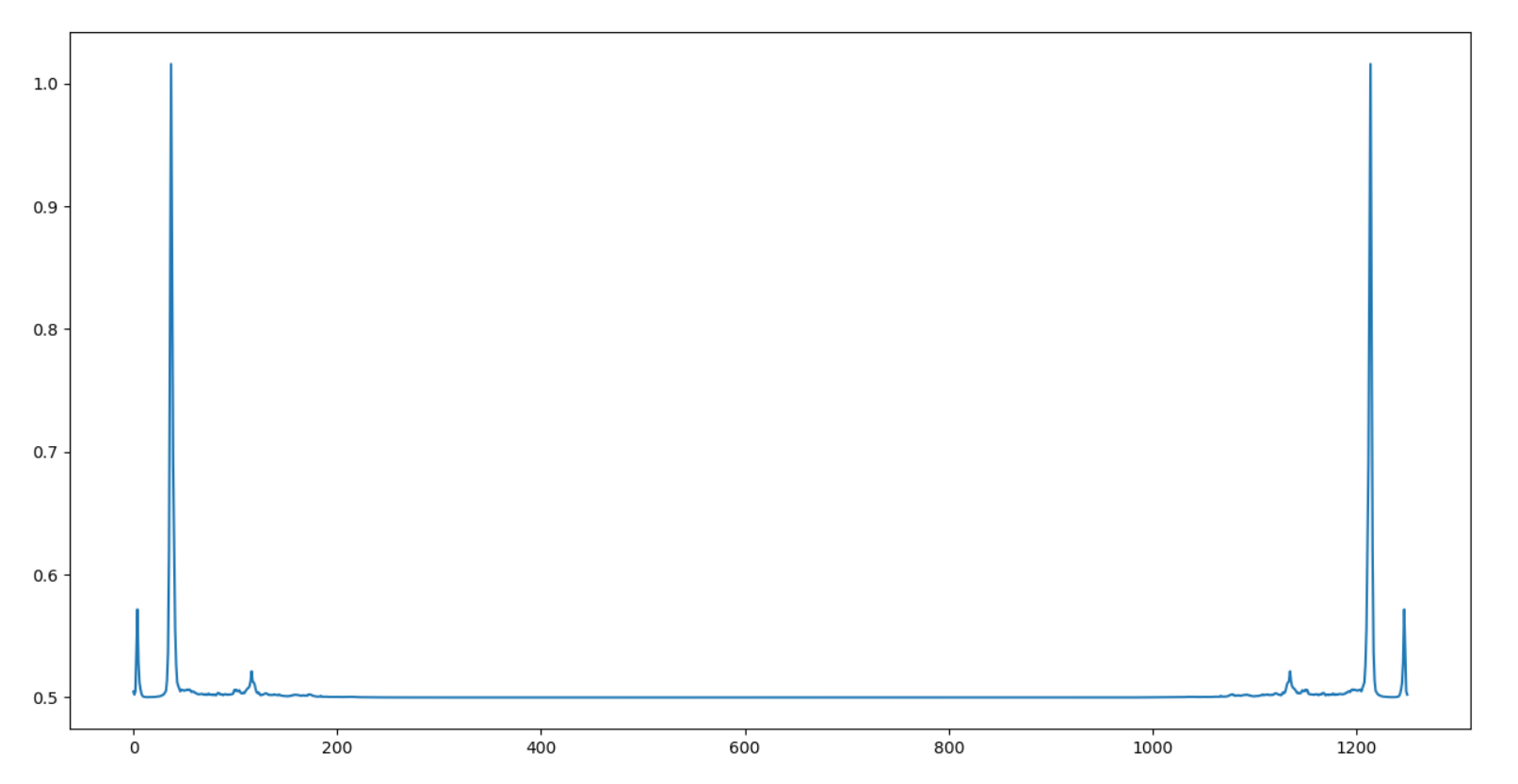
Идея состоит в том, чтобы поэлементно перемножать фурье-образ сигнала на переменную (вектор такой же размерности) в функции потерь, которая будет минимизироваться.

Функция потерь: , где x – сигнал, F – fft, – ifft, V – переменная, которая по сути и есть фильтр, Y – указатель, равный 1, если рассматриваемый сигнал содержит поезд, и 0 в противном случае.

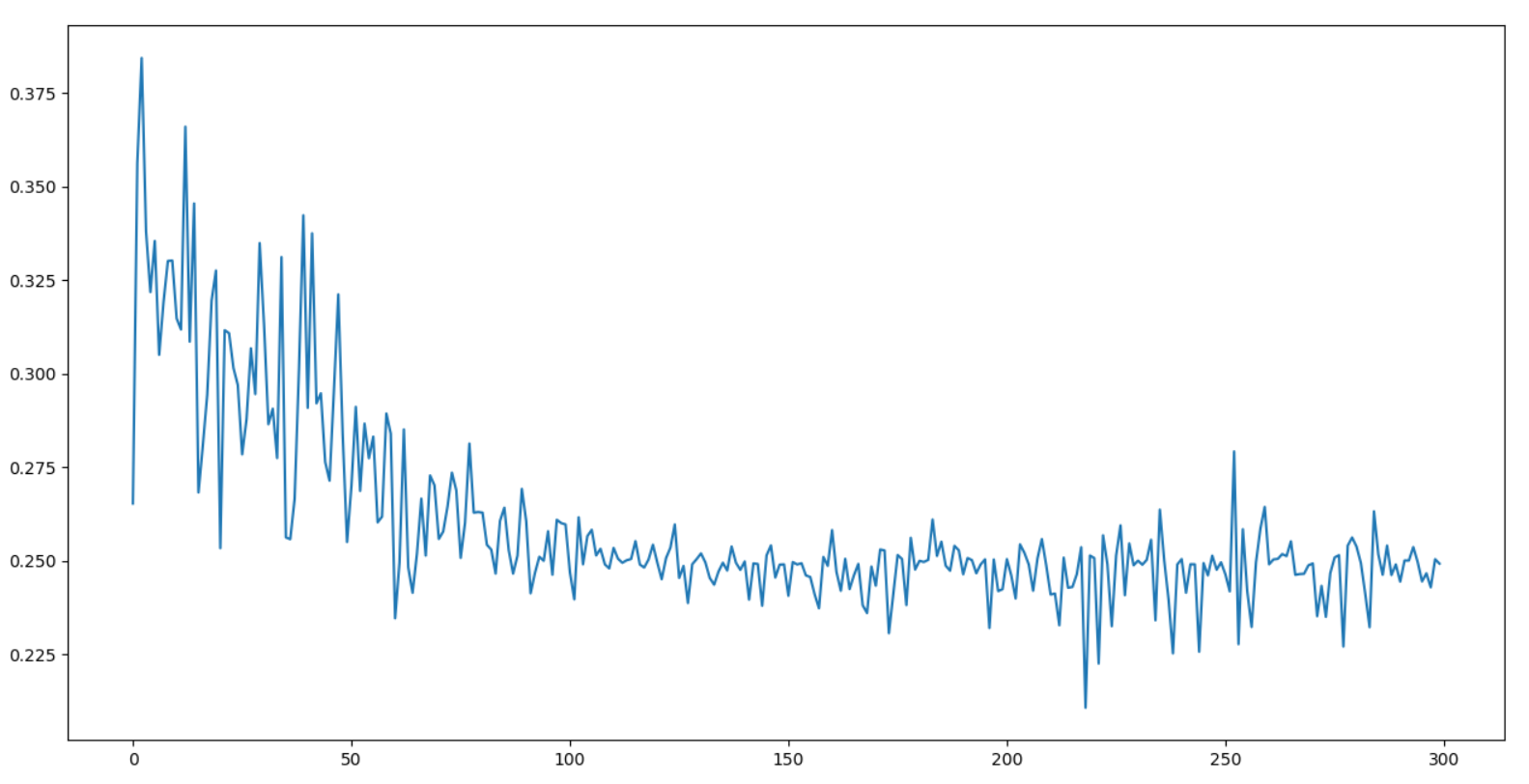
Легко видеть, что если задать все значения V равными к примеру 0.5, функция потерь будет 0.5. Но если V будет действовать как нужный нам фильтр, то норма сигнала с поездом не сильно изменится, значит близко к 1, а к 0, и наоборот, норма сигнала без поезда уменьшится сильно, значит будет ближе к 0, и также будет ближе к 0. Соответственно чем меньше функция потерь, тем лучше фильтр.

Результат для начального значения V = [0.5…0.5] и 300 эпох обучения:

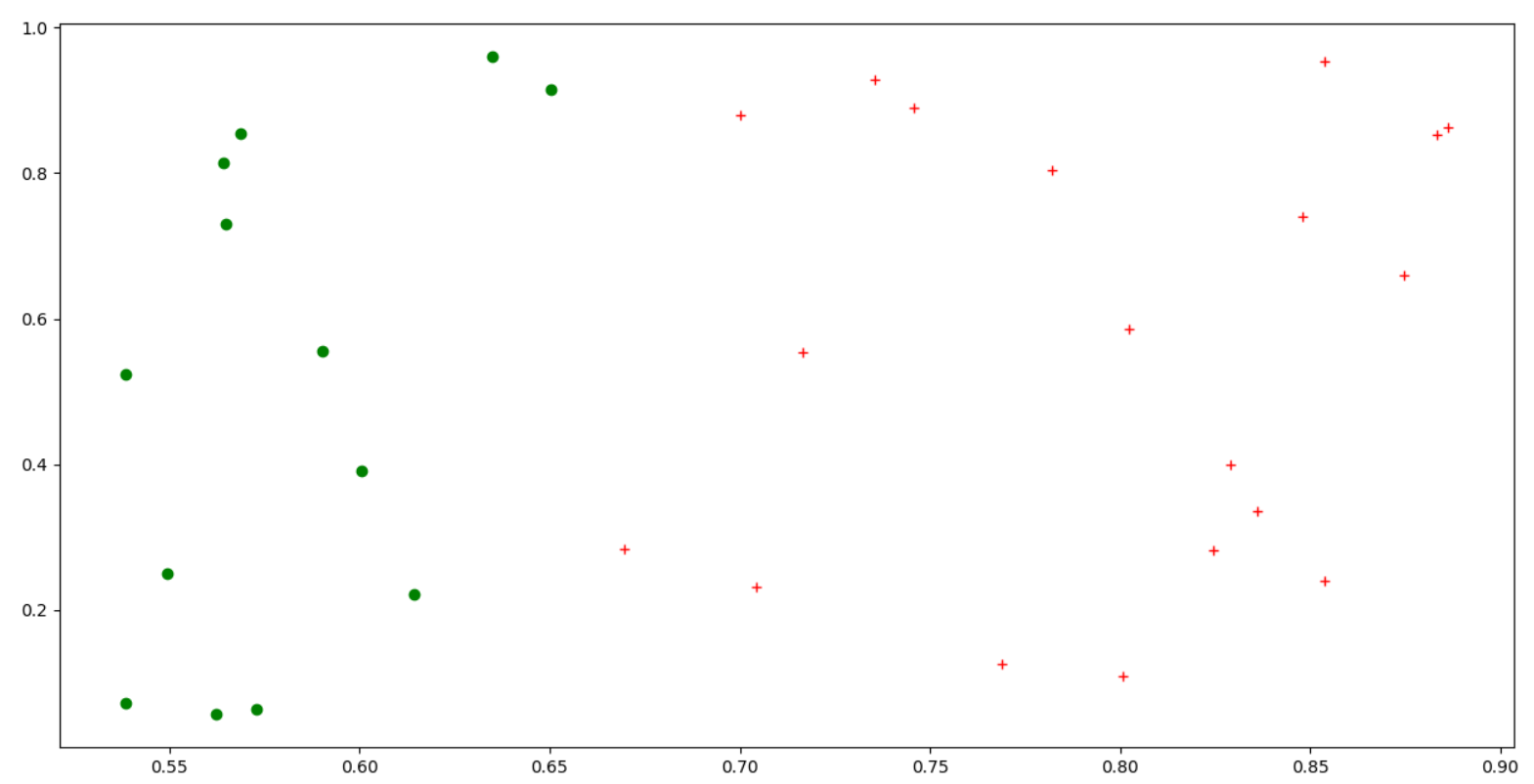
V:



Потери:



Здесь по оси абсцисс показано значение , по оси ординат – случайные числа (чтобы точки не накладывались друг на друга). Круги – данные без поезда, + - с поездом.



Как видно, по значению понятно, содержат данные сигнал от поезда или нет.

Примеры применения фильтра такого типа:

На сигнале со слабым шумом:

|  |  |
| --- | --- |
| До: | После: |
|  |  |

На более зашумленном сигнале:

|  |  |
| --- | --- |
| До: | После (сигнал умножен на 5 т.к. получился довольно слабым): |
|  |  |