

Выпускная квалификационная работа бакалавра на  
тему: «**Веб-сервис для анализа  
сигналов на основе вейвлет-  
нейронной сети**»

**Выполнил:**

Студент группы ИСб-42о  
Дядюшенко Сергей  
Евгеньевич

**Научный руководитель:**

Старший преподаватель  
Строганов Виктор  
Александрович

# Актуальность задачи

В последнее время большую популярность получили технологии, основанные на нейронных сетях. Существует множество продуктов и библиотек для работы с ними.

Однако, до сих пор отсутствуют узконаправленные библиотеки для построения и обучения вейвлет-нейронных сетей.

Многие из имеющихся библиотек не являются достаточно гибкими и расширяемыми.

Не одна из существующих систем никаким образом не доступна для удалённой работы посредством сети интернет. Все известные продукты необходимо разворачивать локально.

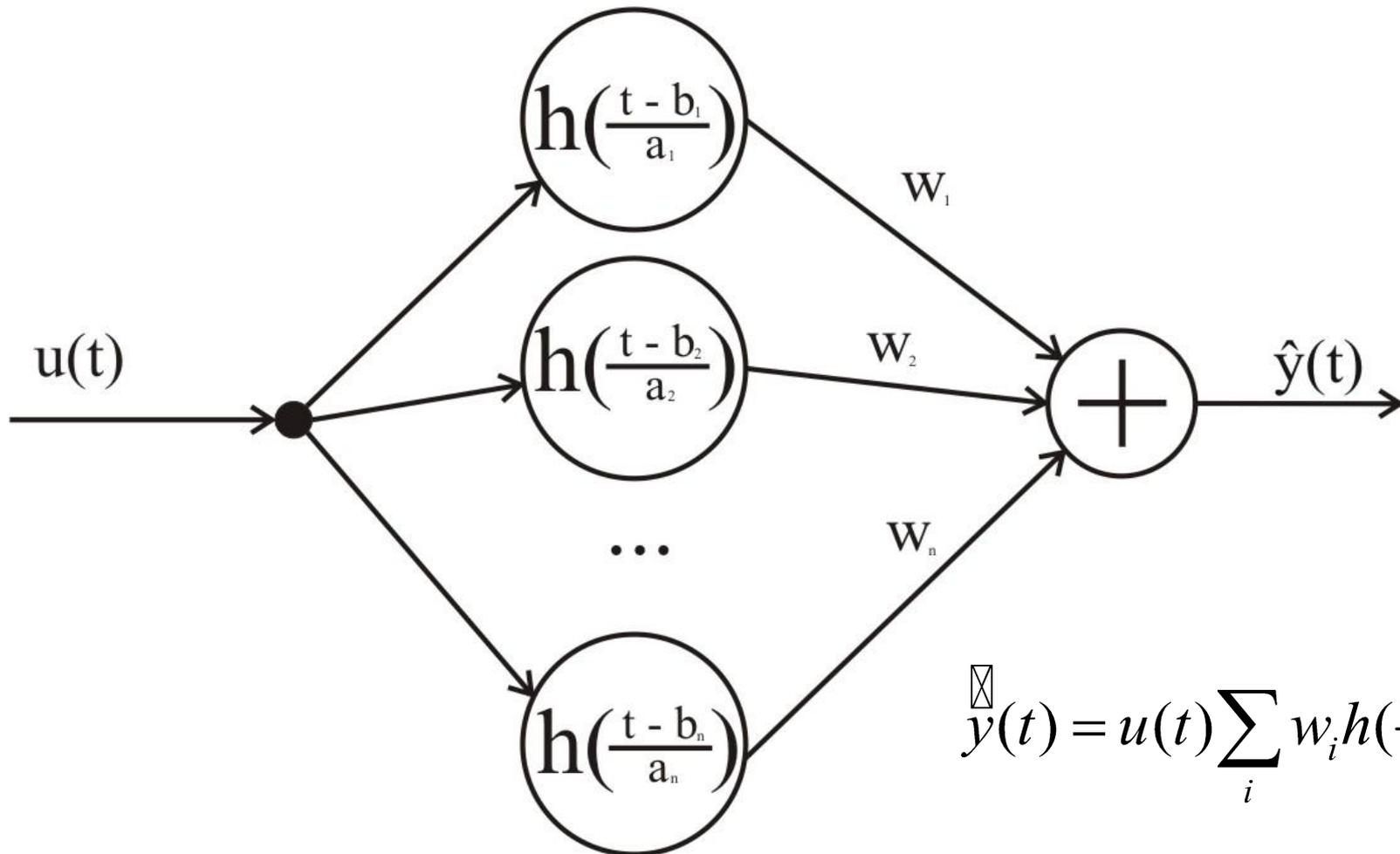
# Цели и задачи

Целью данной работы является проектирование веб-сервиса для анализа сигналов на основе вейвлет-нейронной сети.

Для её достижения необходимо решить следующие задачи:

- изучить математический аппарат вейвлет-нейронных сетей;
- рассмотреть существующие системы, реализующие нейронные сети;
- выполнить проектирование системы;
- осуществить выбор языка программирования и подбор средств разработки;
- реализовать веб-сервис для анализа сигналов на основе вейвлет-нейронной сети;
- разработать автоматические тесты для проверки его работоспособности.

# Вейвлет-нейронная сеть



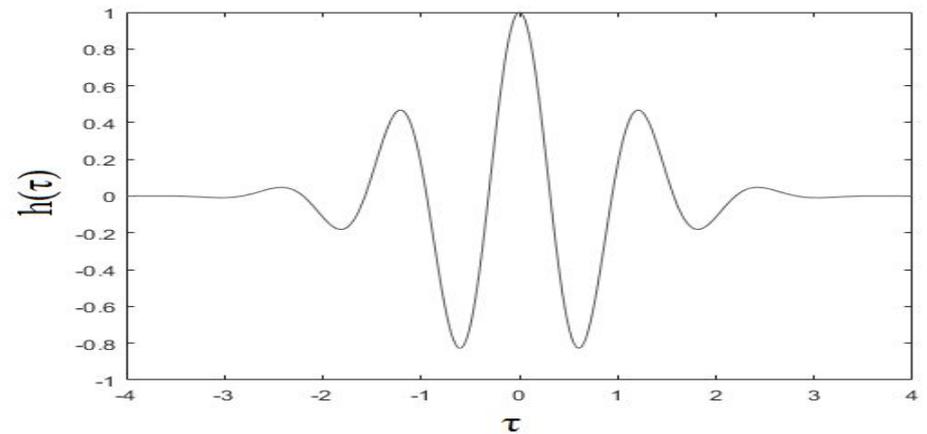
$$\hat{y}(t) = u(t) \sum_i w_i h\left(\frac{t - b_i}{a_i}\right).$$

Вейвлет

Морле

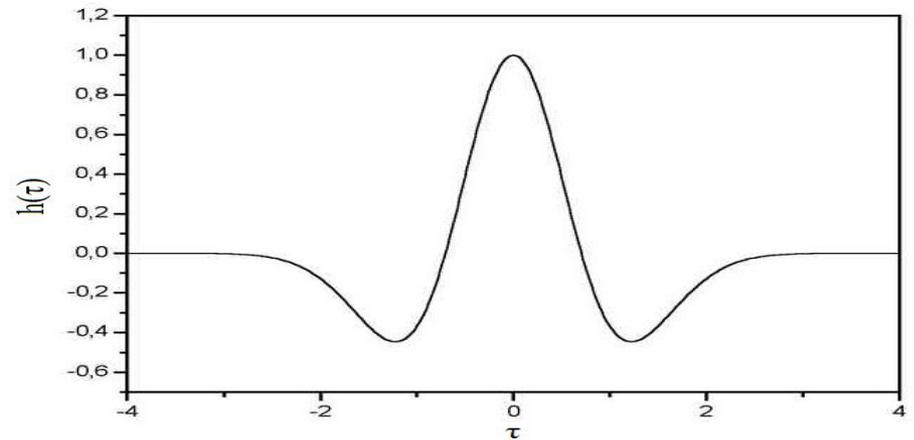
$$y(\tau) = \cos(\omega_0 \tau) \exp(-0.5\tau^2)$$

$$\tau = \frac{t - b_i}{a_i}$$



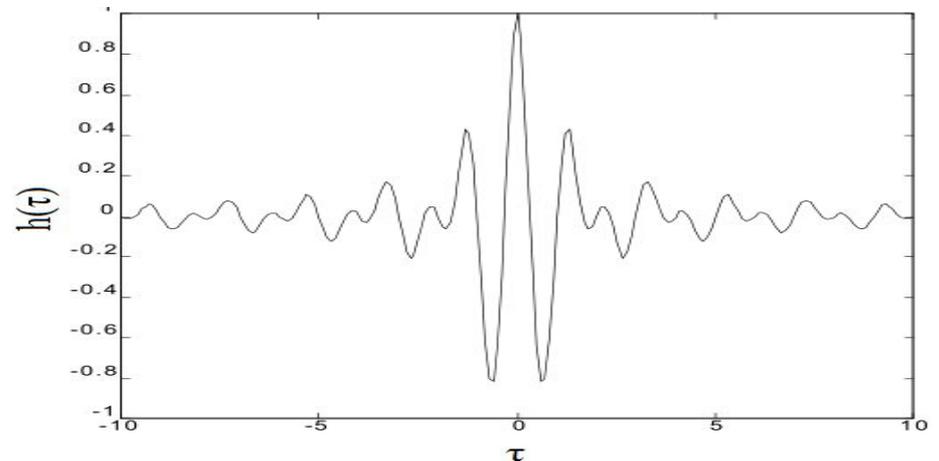
Вейвлет «Мексиканская шляпа»

$$y(\tau) = (1 - \tau^2) \exp\left(-\frac{\tau^2}{2}\right)$$



Вейвлет Шеннона

$$y(\tau) = \frac{\sin 2\pi\tau - \sin \pi\tau}{\pi\tau}$$



# Обучение нейронной сети

Процесс обучения состоит в минимизации функции

ошибки  $E = \frac{1}{2} \sum_{i=1}^T e^2(t)$  где  $e(t) = y(t) - \hat{y}(t)$ .

## Градиентный метод

На каждом шаге происходит изменение параметров сети на вычисляемый градиент

$$\frac{\partial E}{\partial w_i} = - \sum_1^T e(t) h(\tau) u(t)$$

$$\frac{\partial E}{\partial b_i} = - \sum_1^T e(t) w_i \frac{\partial h(\tau)}{\partial b_i}$$

$$\frac{\partial E}{\partial a_i} = - \sum_1^T e(t) w_i \tau \frac{\partial h(\tau)}{\partial b_i}$$

# Инициализация сети

- Простейший метод инициализации

Нейроны делятся на группы с одинаковым коэффициентом масштабирования и коэффициентами сдвига равномерно распределёнными по времени сигнала, веса инициализируются случайными значениями.

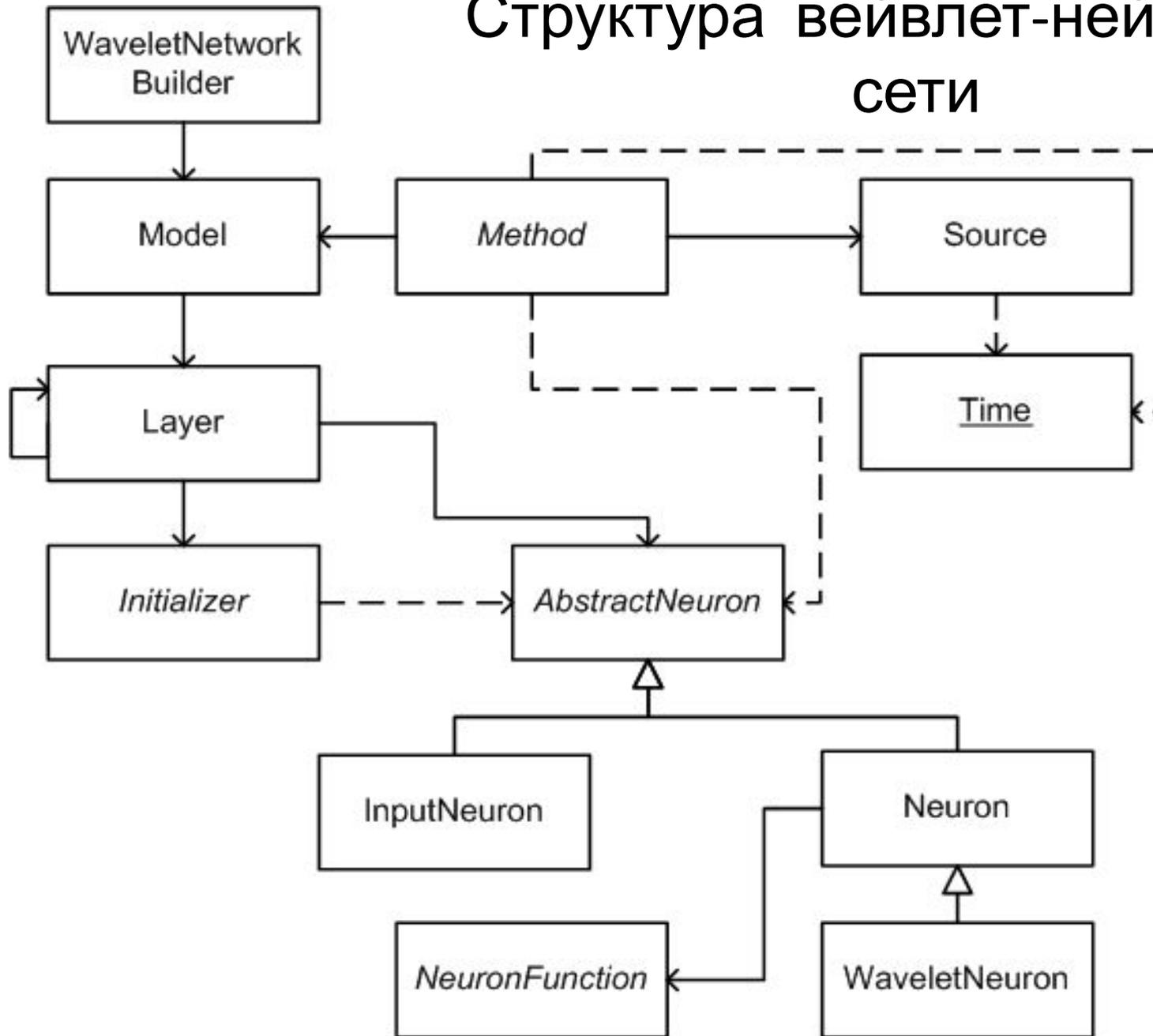
- Инициализация методом остаточного сигнала

Параметры выбираются из сформированной библиотеки подбором наиболее точно описывающих сигнал.

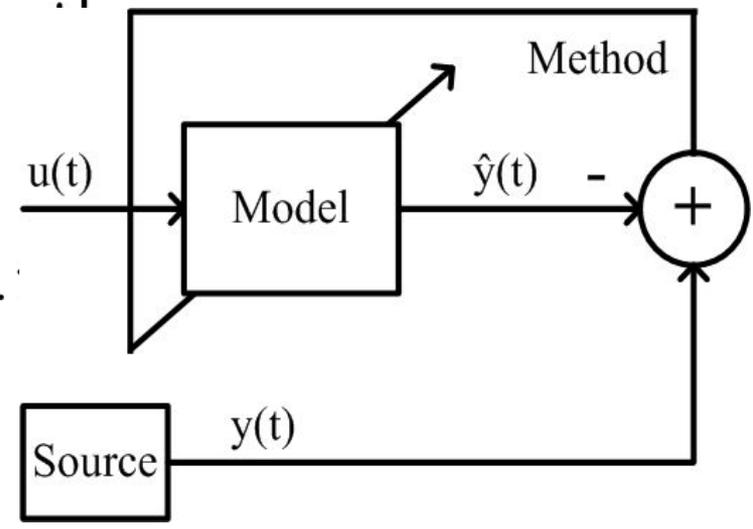
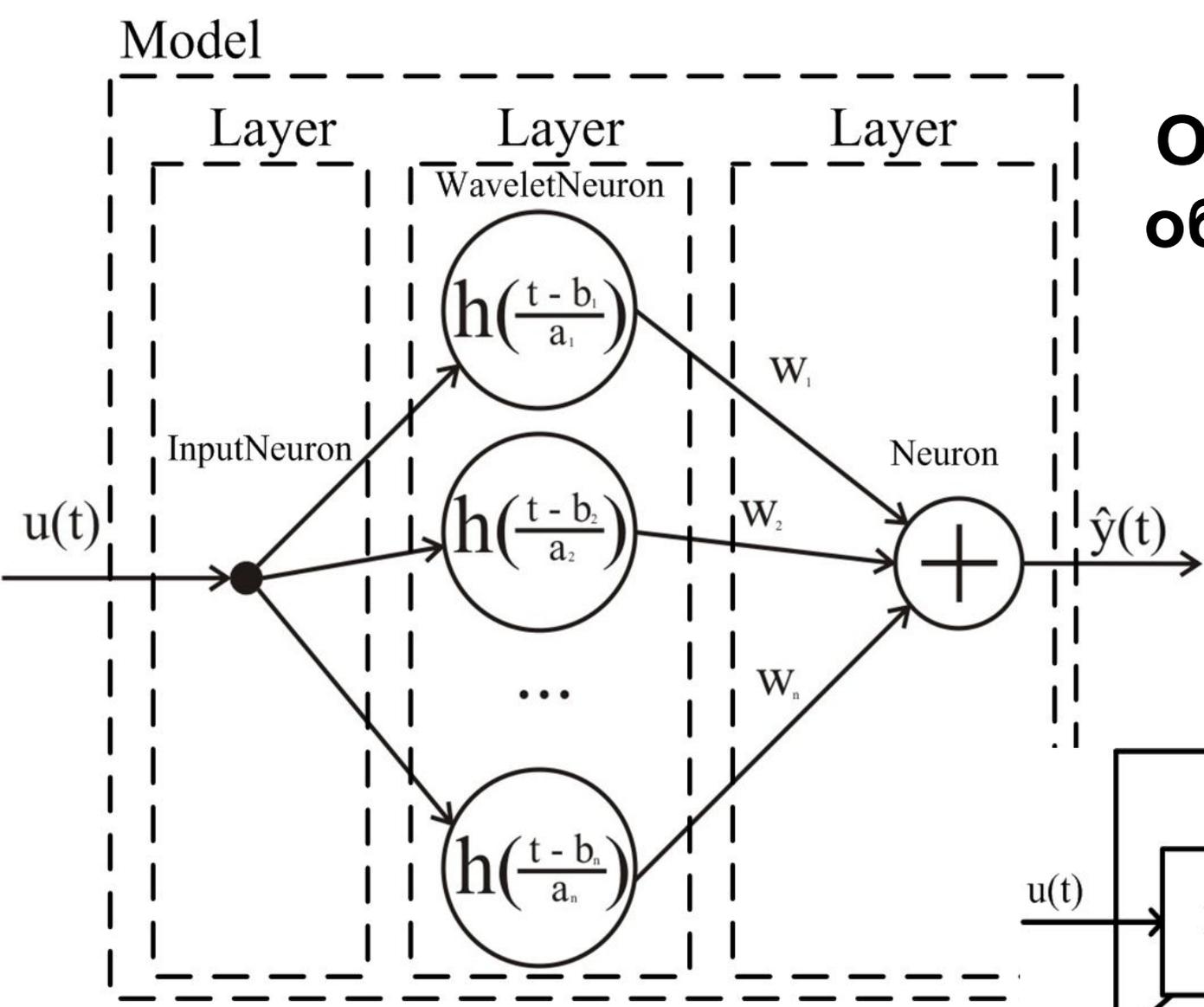
Максимальный коэффициент масштабирования в сети

- задан;

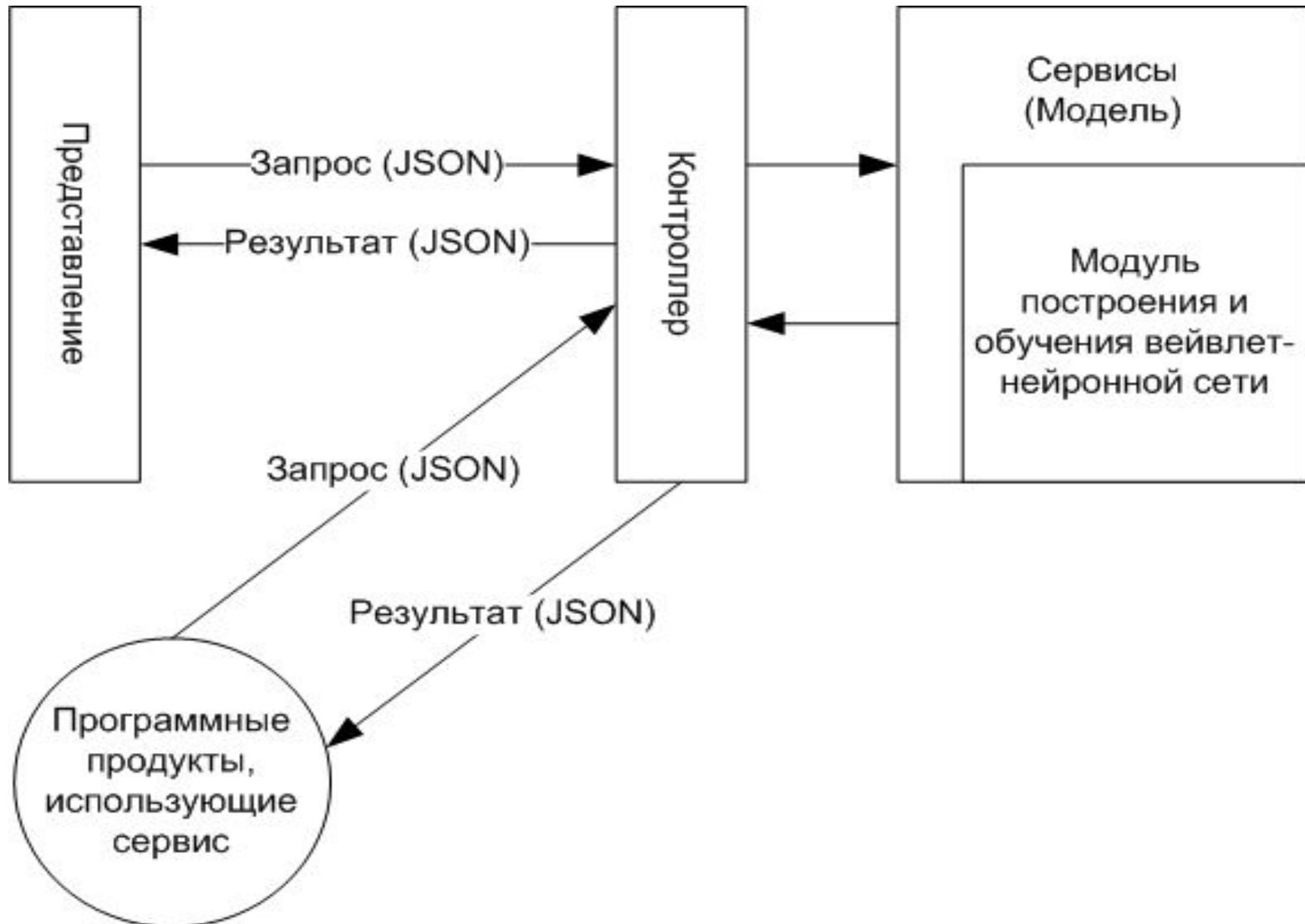
# Структура вейвлет-нейронной сети



# Обозначение обязанностей классов



# Архитектура веб-сервиса



# Использованные технологии



{JSON}



JUnit



Chart.js

# Пример запроса и ответа

## Сервис построения и обучения вейвлет-нейронной сети

```
{
"method": "gradient",
"aMu": 0.00000000000000015,
"bMu": 0.00000000000000001,
"wMu": 0.0005,
"initialization": "standart",
"dilation": 0.002,
"groupSize": 4,
"func": "morlet",
"iterationsCount": 200,
"error": 0.1,
"time": 1000,
"neuronsCount": 16,
"fullOutput": true,
"source": [
  {"t": 0.0, "y": -0.08078},
  {"t": 2.26757E-5, "y": -0.0323792},
  ...
]
}
```

```
{
"parameters": {
  "a": [
    4.0E-4,
    ...
  ] ,
  "b": [
    0.002699,
    ...
  ],
  "w": [
    0.388434,
    ...
  ]
},
"output": [
  -0.059953,
  ...
]
}
```

# Пример запроса и ответа

Сервис получения выхода сети по заданным параметрам

```
{
  "parameters": {
    "a": [
      4.0E-4,
      ...
    ],
    "b": [
      0.002699,
      ...
    ],
    "w": [
      0.388434,
      ...
    ]
  },
  "func": "morlet",
  "t": [
    0,
    2.267573E-5,
    ...
  ]
}
```

# Графический интерфейс СИСТЕМЫ

Method

Gradient method

Learning coefficient for dilation

1.5e-14

Learning coefficient for translation

1e-15

Learning coefficient for weight

0.0005

Initialization

More stable

Maximal dilation

0.002

Function

Morlet

Error

0.1

Iterations count

200

Time to processing (milliseconds)

0 means processing until will be achieved error or iterations count.

0

Neurons count

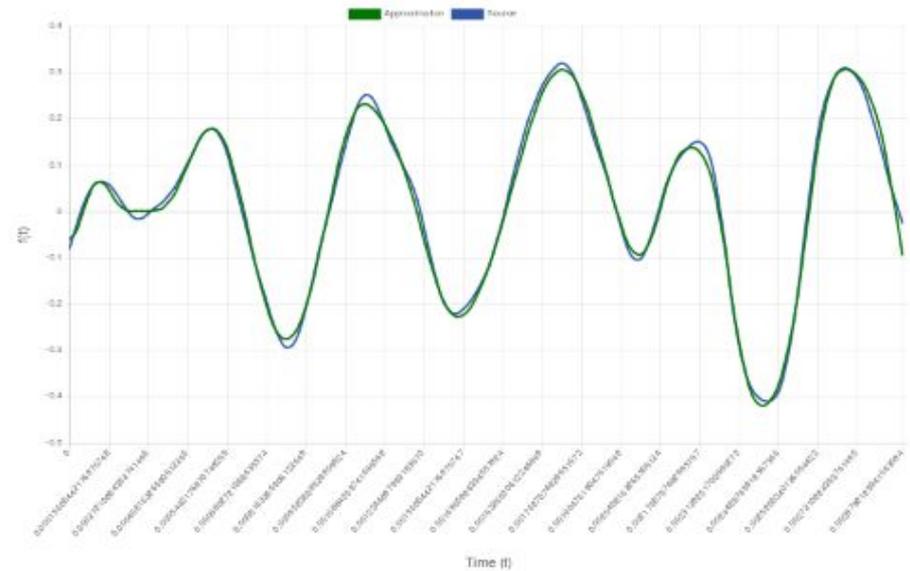
10

Values

```

0.002630385487528345,0.2732849
0.002653061224488796,0.2968445
0.0026757369614512473,0.3002744
0.002698412698412698,0.3044128
0.002721084353741495,0.2896118
0.0027437641723256007,0.261261
0.00276643990297052,0.2200421
0.002789115640286033,0.1714478
0.0028117913832189546,0.1174011
0.002834467120181406,0.0638937
0.002857142857142857,0.0158081
    
```

Submit

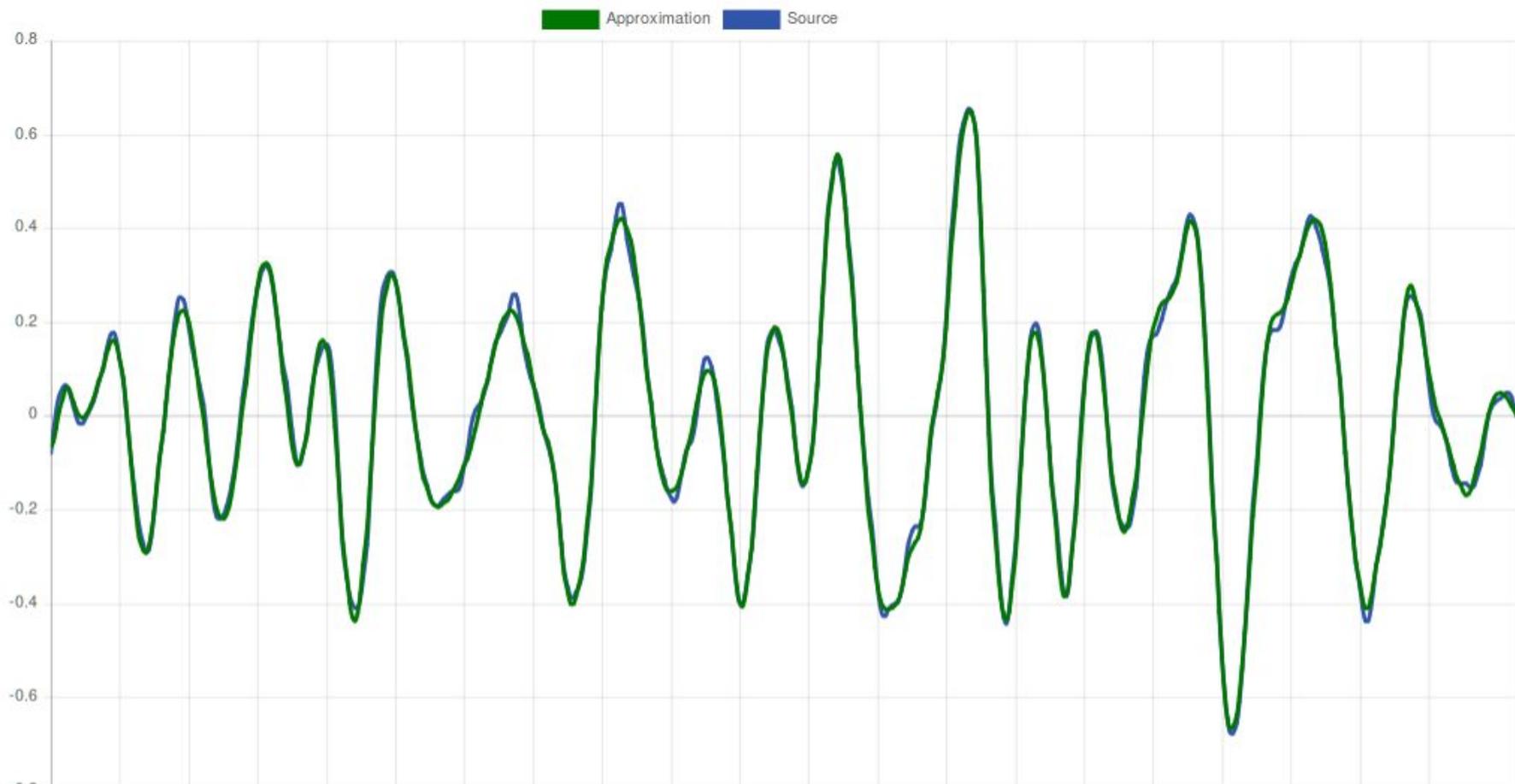


Network Parameters

Number	A	B	W
1	0.0004	0.0326982593119727883	0.2884347420622217
2	0.0004	0.0010798119727881157	0.31675649839726415
3	0.001	0.0017988996213151628	0.103197187363118
4	0.0003333333333333333	0.00197875283446712	-0.11709412618939887
5	0.0003333333333333333	0.0008969433108578964	0.10341159479757305
6	0.0009999999999999999	0.0021598639425762313	-0.18132772189462947
7	0.00025	0.0022388528877097585	-0.06771296252875279
8	0.002	0.0010798119727881157	0.029007012696333185
9	0.00025	0.0002997732403032295	-0.0614384928222233
10	0.0003333333333333333	0	-0.061195959250036135
11	0.0003333333333333333	0.0012598230349336325	-0.02900677959393947
12	0.0001999999999999999	0.0326982593119727883	-0.03133620102478826
13	0.0001999999999999999	0.002511894128994127	-0.03408480641403049
14	0.0003333333333333333	0	0.02463671527934986
15	0.0003333333333333333	0.0021598639425762313	-0.02980650799921572
16	0.0001999999999999999	0.0010798119727881157	-0.022303449432545446

Try again

# Результаты работы



Результаты автоматического тестирования:

```
[the-neural-network] $ test  
[info] Passed: Total 21, Failed 0, Errors 0, Passed 21  
[success] Total time: 16 s, completed 31.05.2016 23:05:41  
[the-neural-network] $
```

# Выводы

Таким образом была достигнута поставленная цель – разработка веб-сервиса аппроксимации сигналов на основе вейвлет-нейронной сети.

Система была разработана модульно на основе принципов ООП, следовательно её можно усовершенствовать и расширять.

## **Направления дальнейшего развития системы:**

- включение дополнительных вейвлет-функций;
- реализация методов инициализации сети;
- реализация методов обучения сети;
- оптимизация системы.

**Спасибо за внимание**