

Московский государственный технический университет имени
Н.Э.Баумана



Нейросетевой подход к обратной задаче генерации поверхностных волн

Автор:

Сычугина А.С., студентка группы ФН1-47

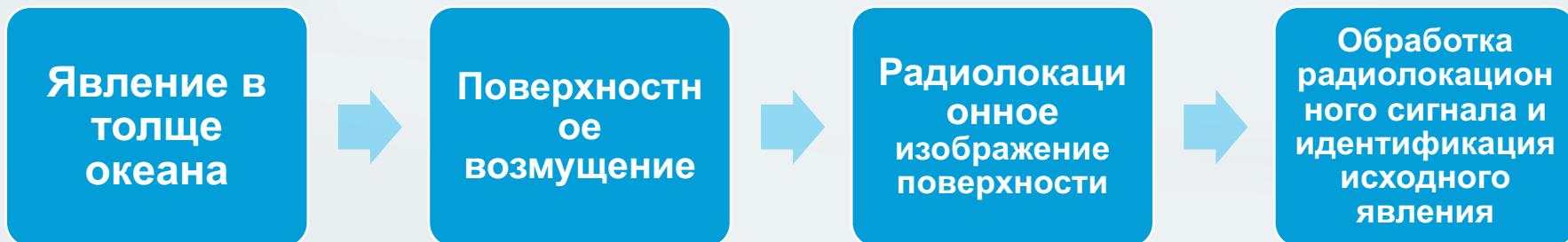
Научный руководитель: Савин А.С., доктор ф.-м.наук, профессор

Москва, 2018



Актуальность

- анализ динамики поверхности и толщи морской среды
- создание нового научно-технического направления – **компьютерной радиотомографии** морской среды



модель информационного тракта радиотомографического комплекса



Цель и задачи исследования

Цель:

Разработать программный комплекс, предназначенный для определения параметров источника возмущений по данным о вызываемых им поверхностных волнах

Решаемые задачи:

1. Применить методы машинного обучения, в частности нейросетевой подход для решения обратной задачи генерации поверхностных волн
2. Построить нейросетевую модель для предсказания глубины и скорости источника возмущения в толще жидкости
3. Реализовать алгоритм обучения построенной модели
4. Провести численные эксперименты, сравнить полученные результаты с данными, найденными в ходе лабораторного эксперимента



Постановка задачи в рамках концепции нейронных сетей

Пусть задана область $\Omega : \vec{r} = (x_1, x_2, x_3) \in \Omega \subset \mathbb{R}^3$ и заданы два множества точек:

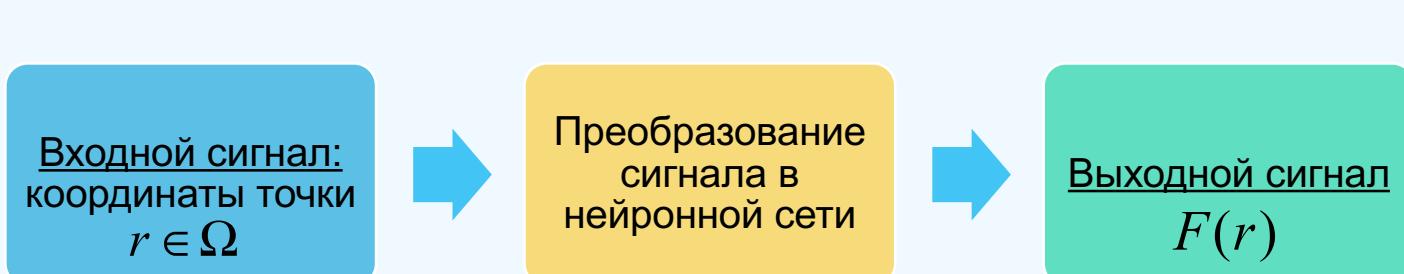
$Q_N = \left\{ r_n \right\}_{n=1}^N \in \Omega$ - **обучающее множество**

$Q_M = \left\{ r_m \right\}_{m=1}^M \in \Omega$ - **целевое множество**

$f(Q_N)$ - значения функции в точках обучающего множества

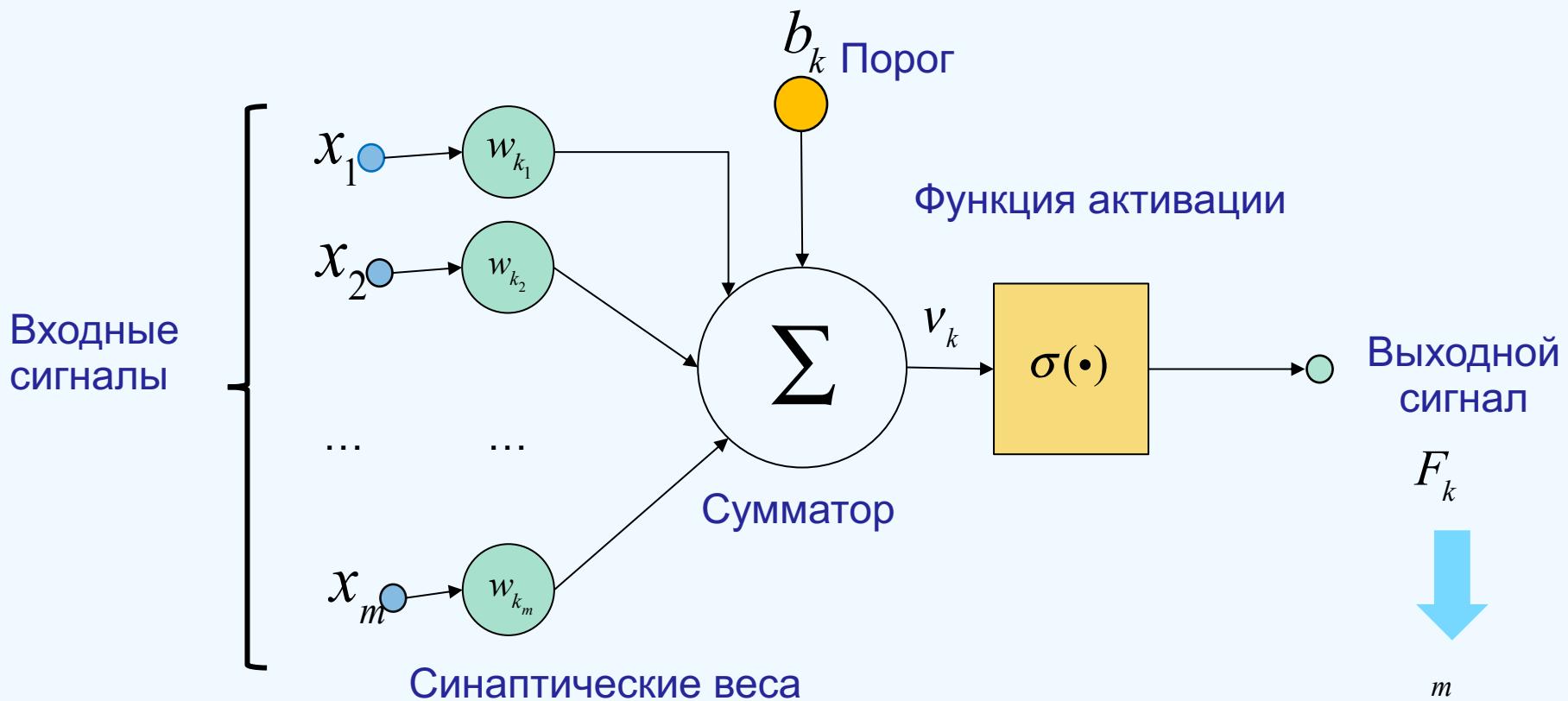
$F(r)$ - аппроксимирующая функция, которая строится с помощью нейронной сети как непрерывное отображение

$$F : \Omega \in R^3 \rightarrow R$$





Понятие формального нейрона

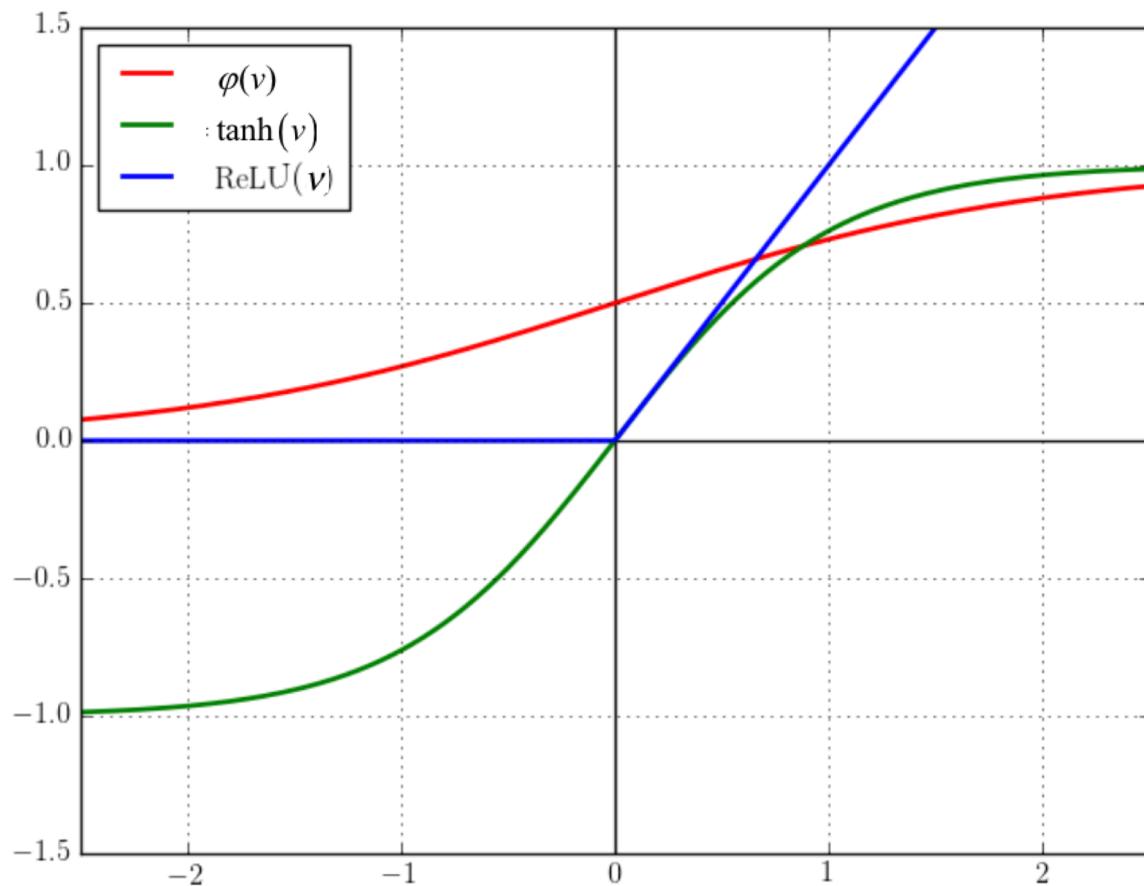


$$u_k = \sum_{j=1}^m w_{kj} x_j,$$

$$F_k = \varphi \left(\sum_{i=1}^m w_{ki} x_i + b_k \right)$$



Выбор функции активации



Примеры функций активации

$$\phi(v) = \frac{1}{1 + \exp(-av)}$$

$$\phi(v) = \tanh(v).$$

$$\phi(v) = \max(0, v)$$

...

Используем:

- На всех скрытых слоях - гиперболический тангенс
- На выходном слое - линейная функция

a – параметр наклона

Выражение для профиля свободной границы потока

$$S(x) = \frac{\Gamma}{\pi V} \int_{-\infty}^x \frac{(t - x_0) \cos v(t-x) + y_0 \sin v(t-x)}{(t - x_0)^2 + y_0^2} dt.$$

$$\frac{d^2 S}{dx^2} + v^2 S = \frac{\Gamma}{\pi V} \frac{1}{(x - x_0)^2 + y_0^2} \left[\frac{2y_0^2}{(x - x_0)^2 + y_0^2} - vy_0 - 1 \right]$$

Обтекание точечного вихря

интенсивности Γ ,

локализованного в точке: $z_0 = -ih$

$$\frac{d^2 S}{dx^2} + v^2 S = \frac{\Gamma}{\pi V} \cdot \frac{1}{h^2 + x^2} \cdot \left(\frac{h^2 - x^2}{h^2 + x^2} + vh \right)$$

$$S(0) = \frac{\Gamma}{\pi V} e^{-vh} Ei(vh), \quad \frac{dS}{dx}(0) = \frac{\Gamma v}{V} e^{-vh}$$

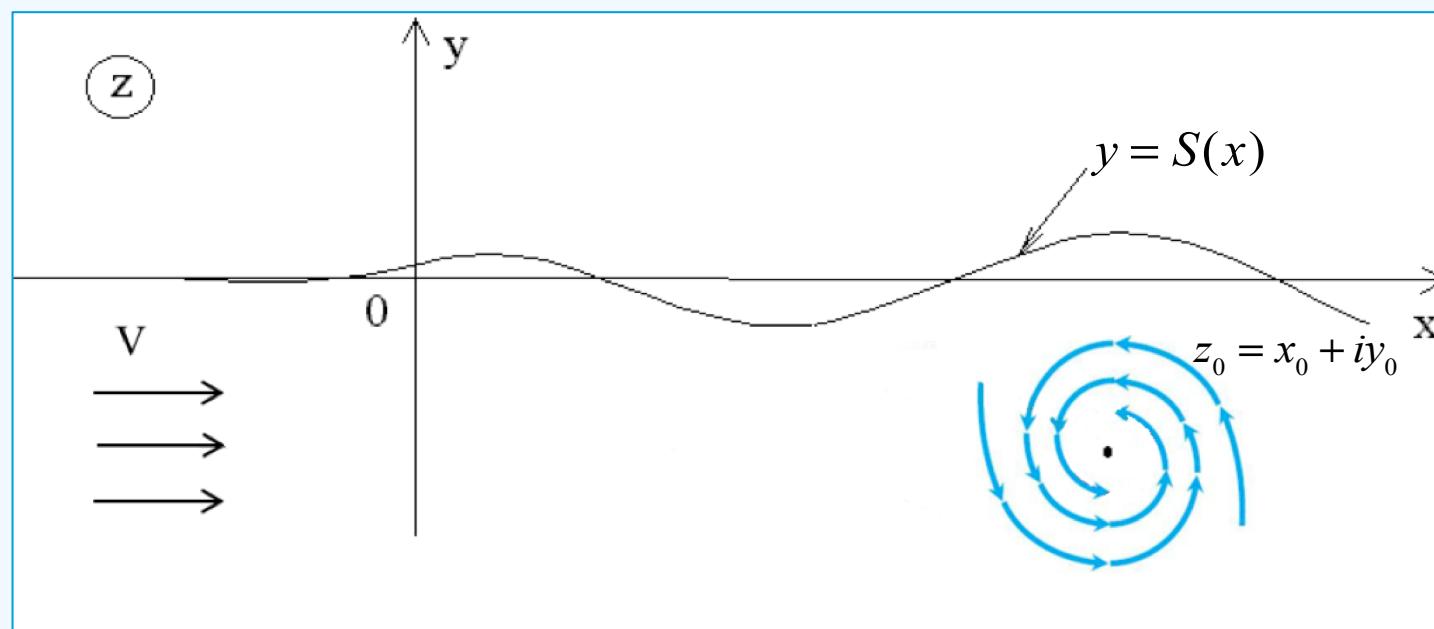
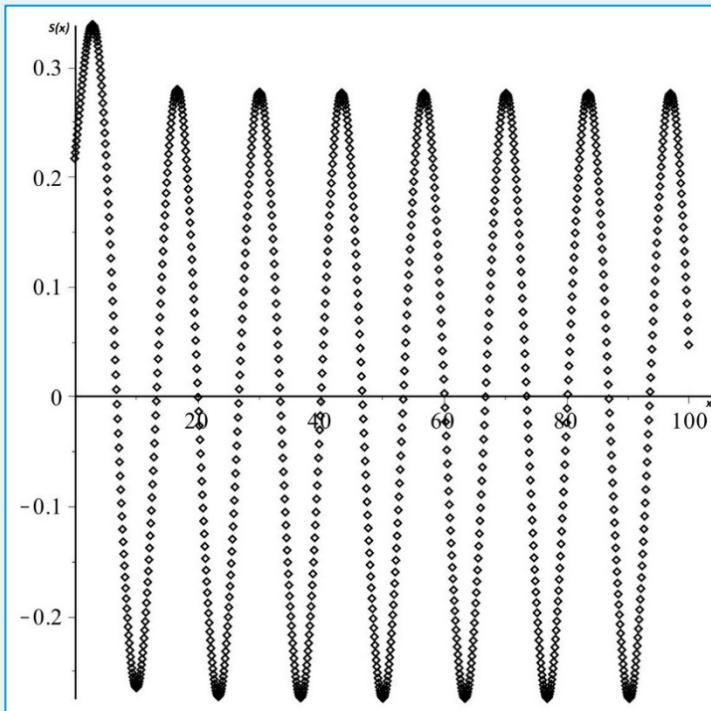
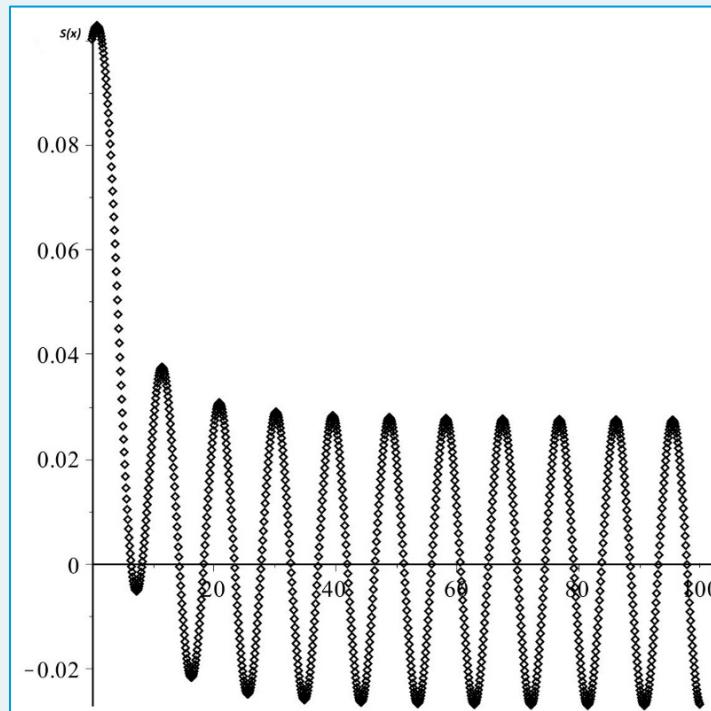


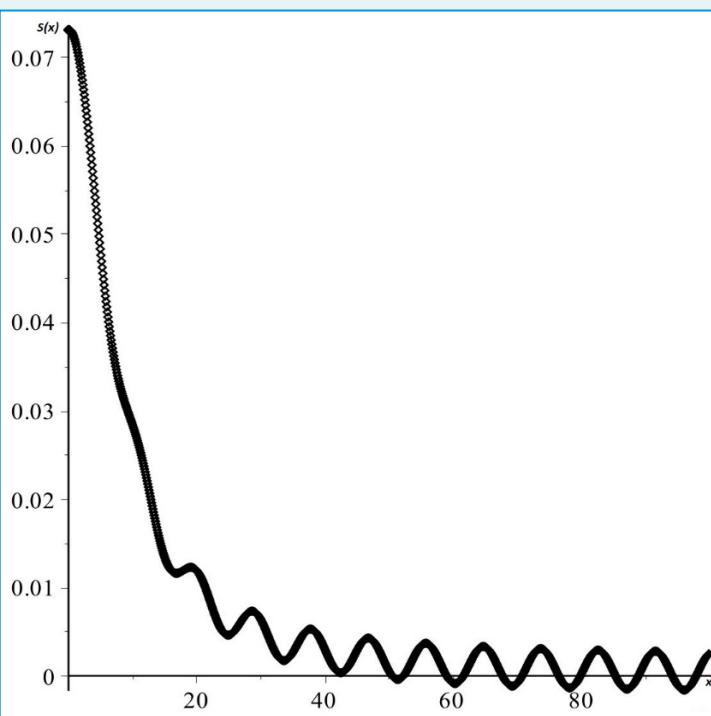
Схема обтекания точечного вихря потоком со свободной границей (пример)



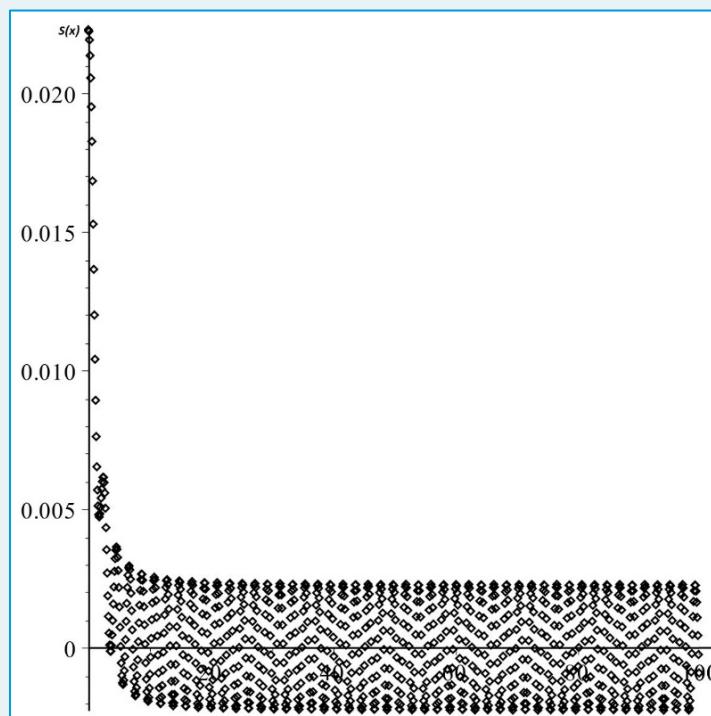
$\Gamma = 4.389; h = 3.762; \nu = 0.469; V = 4.569$



$\Gamma = 3.123; h = 5.624; \nu = 0.674; V = 3.813$



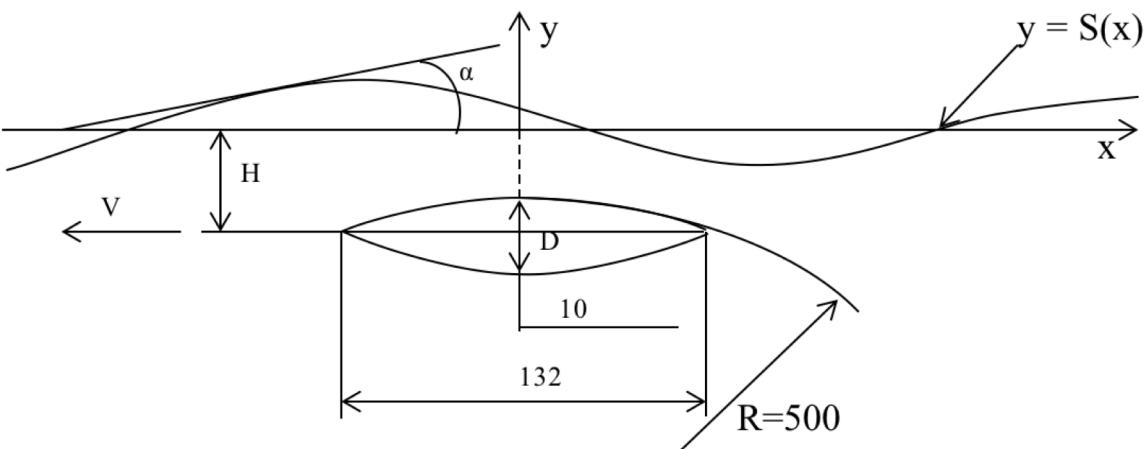
$\Gamma = 2.441; h = 9.064; \nu = 0.857; V = 3.381$



$\Gamma = 0.436; h = 1.565; \nu = 3.052; V = 1.792$



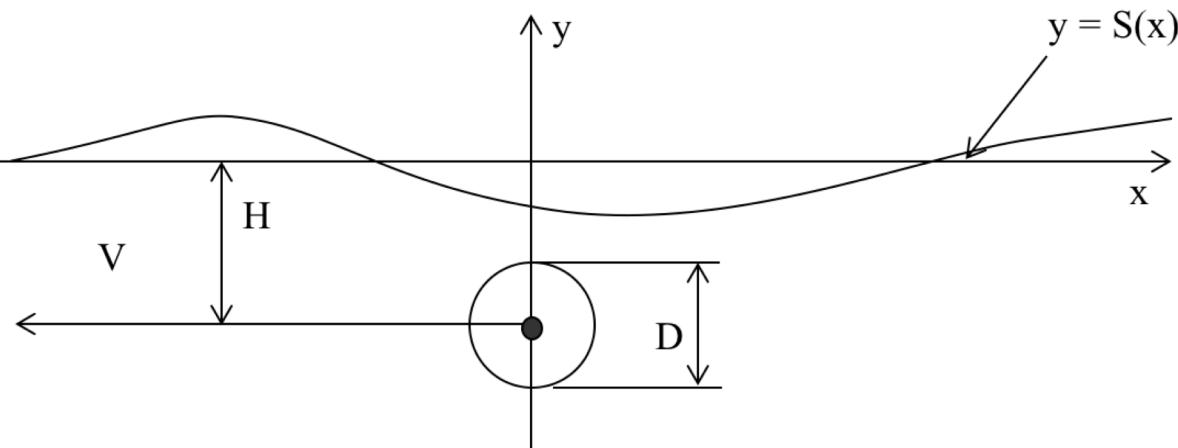
Получение экспериментальных данных



Движение крылового профиля

$H=175\text{мм}$

V различны



Движение кругового цилиндра

$D=6\text{мм}, D=20\text{мм}$

$H = 66-133 \text{ мм}$

$V=280-1056 \text{ мм/с}$

* Эксперимент был проведен в Институте проблем механики РАН



Предварительная обработка данных. Генерация обучающей выборки

Применение
спектрального
анализа к данным,
полученным путем
физического
эксперимента

- для выявления характерных признаков
- для сокращения обучающей выборки

Формирование
выборки большего
размера путем
зашумления (5%)

- для хорошего обобщения
- избежать переобучения

Нормирование
данных

- для проведения процесса обучения по вариативности факторов

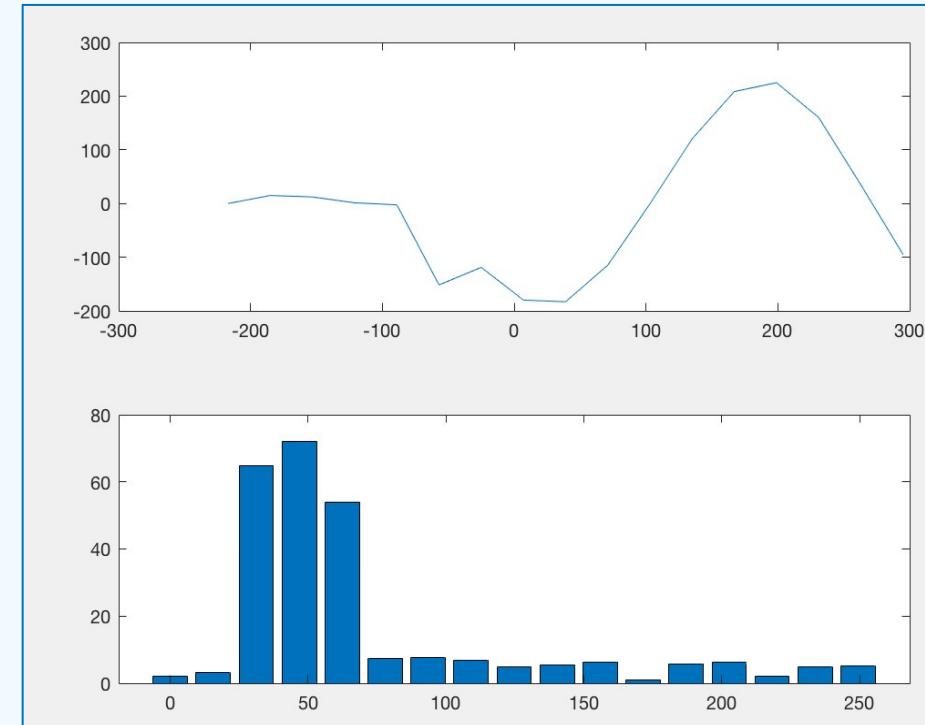
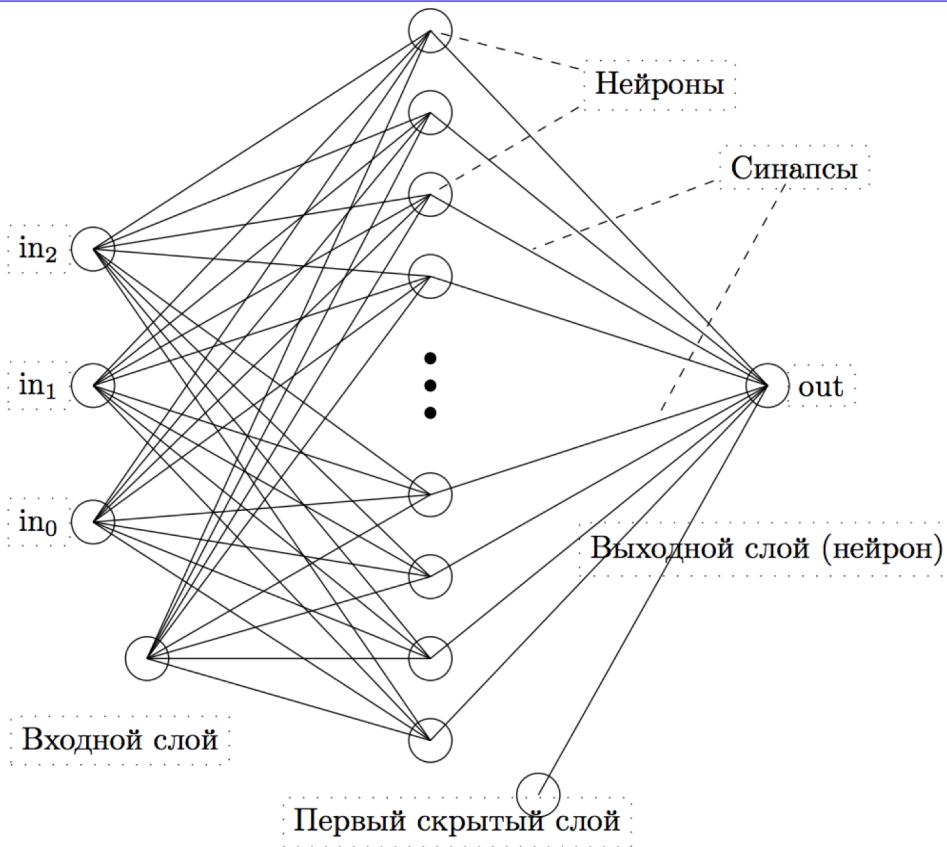


График полученного спектра Фурье
для крылового профиля



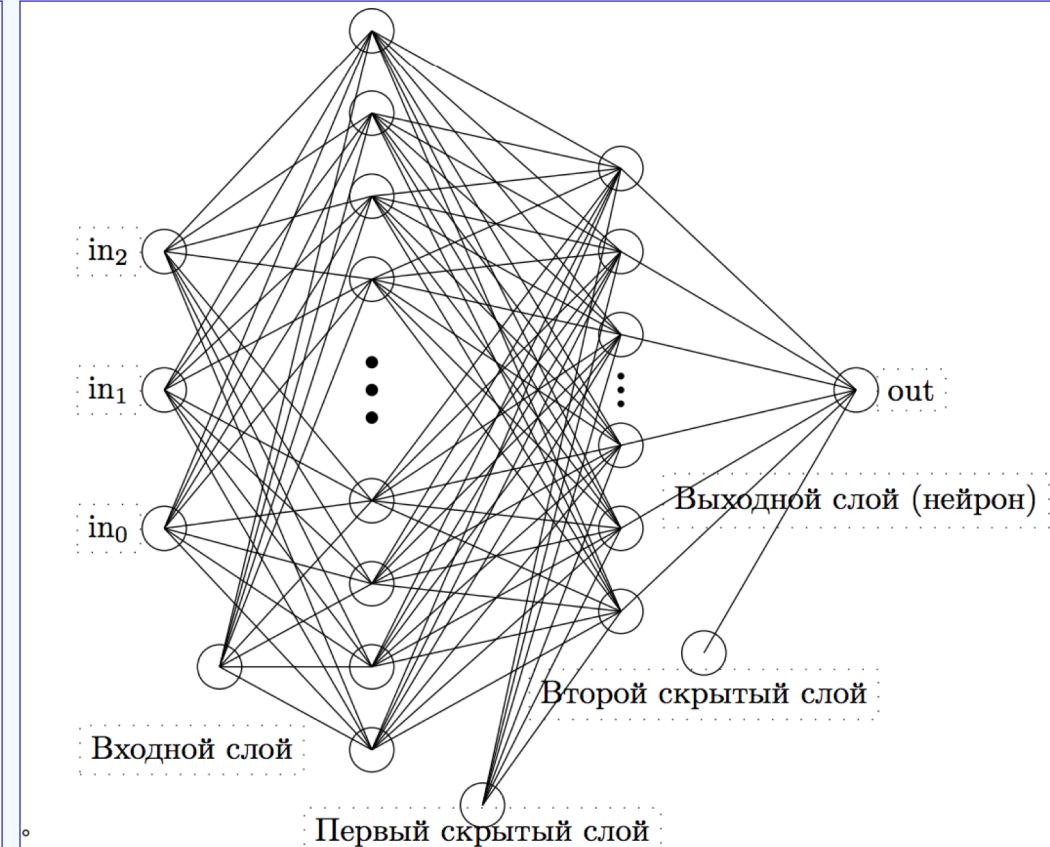
Выбор оптимальной архитектуры нейронной сети

3-20-1(tanh) – 3 входных, 20 скрытых и 1 выходной нейрон с функцией активации гиперболический тангенс



полносвязная нейронная сеть с одним скрытым слоем

Полносвязная нейронная сеть – каждый нейрон передает свой выходной сигнал на вход остальных нейронов, а также самому себе



полносвязная нейронная сеть с двумя скрытыми слоями



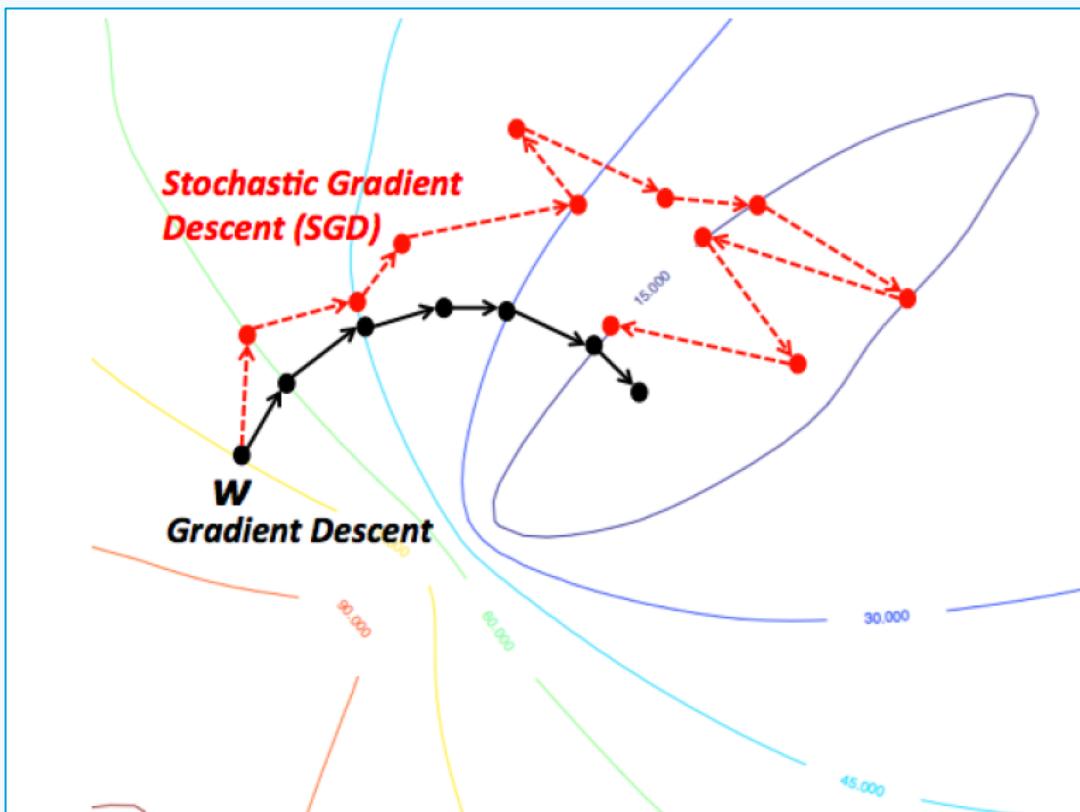
Метод SGD для оптимизации обучения персептрона

Цель обучения – минимизация выходной ошибки сети на обучающем множестве

$$E = \frac{1}{2p} \sum_{i=1}^p (d_i - y_i)^2$$

y_i – фактический выходной сигнал сети
 d_i – желаемый отклик сети

Процесс обучения – подстройка весов нейронов



Линии уровня. Изменения весов

Градиентный метод численной минимизации:

$$Q(w) = \sum_{i=1}^l L(\langle w, x_i \rangle y_i) \rightarrow \min_w,$$

$$w^{(t+1)} = w^{(t)} - \eta \cdot \nabla Q(w^{(t)}),$$

$$\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=0}^n,$$

η – градиентный шаг (скорость обучения)

Обновление весов:

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^l L(\langle w^{(t)}, x_i \rangle x_i y_i).$$



Мониторинг состояния нейронной сети [1]

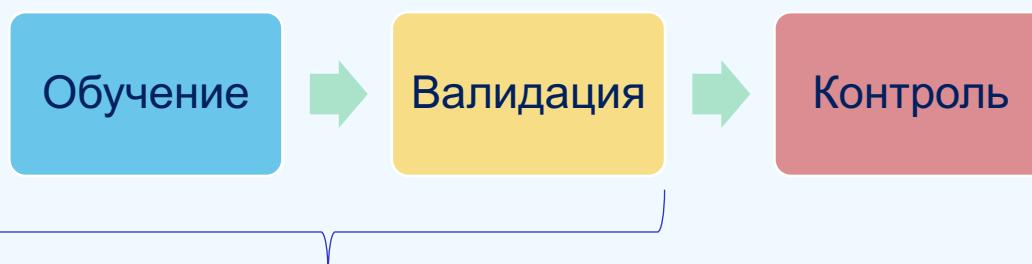
Гиперпараметры — значения, которые нужно подбирать вручную и зачастую методом проб и ошибок при обучении

Эпоха - одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и проверку качества обучения на контрольном множестве

Примеры гиперпараметров

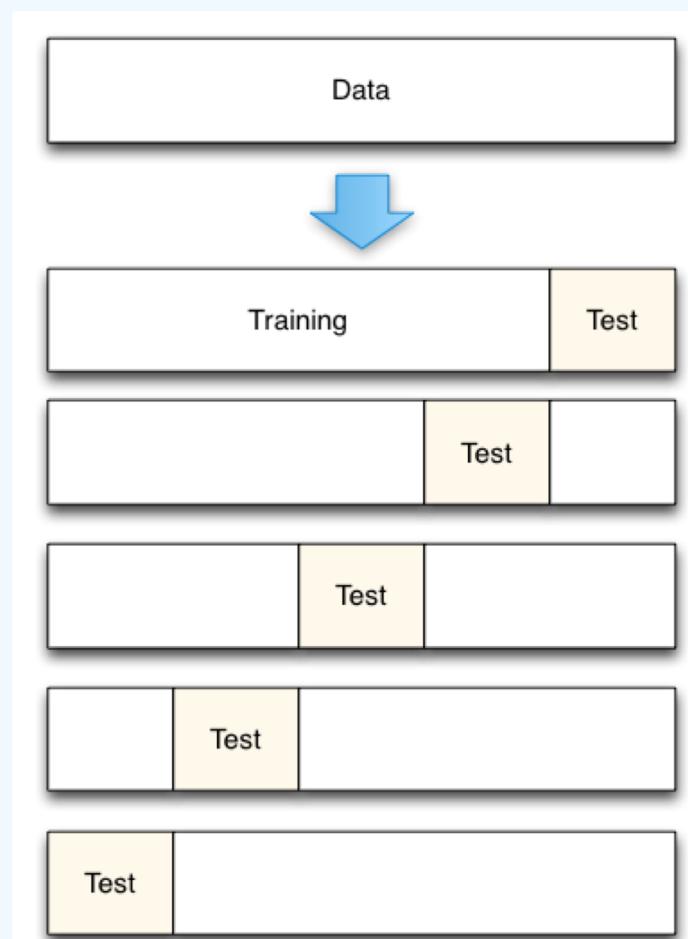
- Момент и скорость обучения
- Количество скрытых слоев
- Количество нейронов в каждом слое
- Количество эпох

Схема обучения



Кросс-валидация (k-fold cross-validation, **k-fold-cv**)

Обычная валидация (simple validation, **s-val**)





Выбранные метрики качества (критерии качества в задаче регрессии)

MSE – среднеквадратичная ошибка

$$MSE(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2$$

- легко минимизировать
- сильнее штрафует за большие отклонения

MAE – средняя абсолютная ошибка

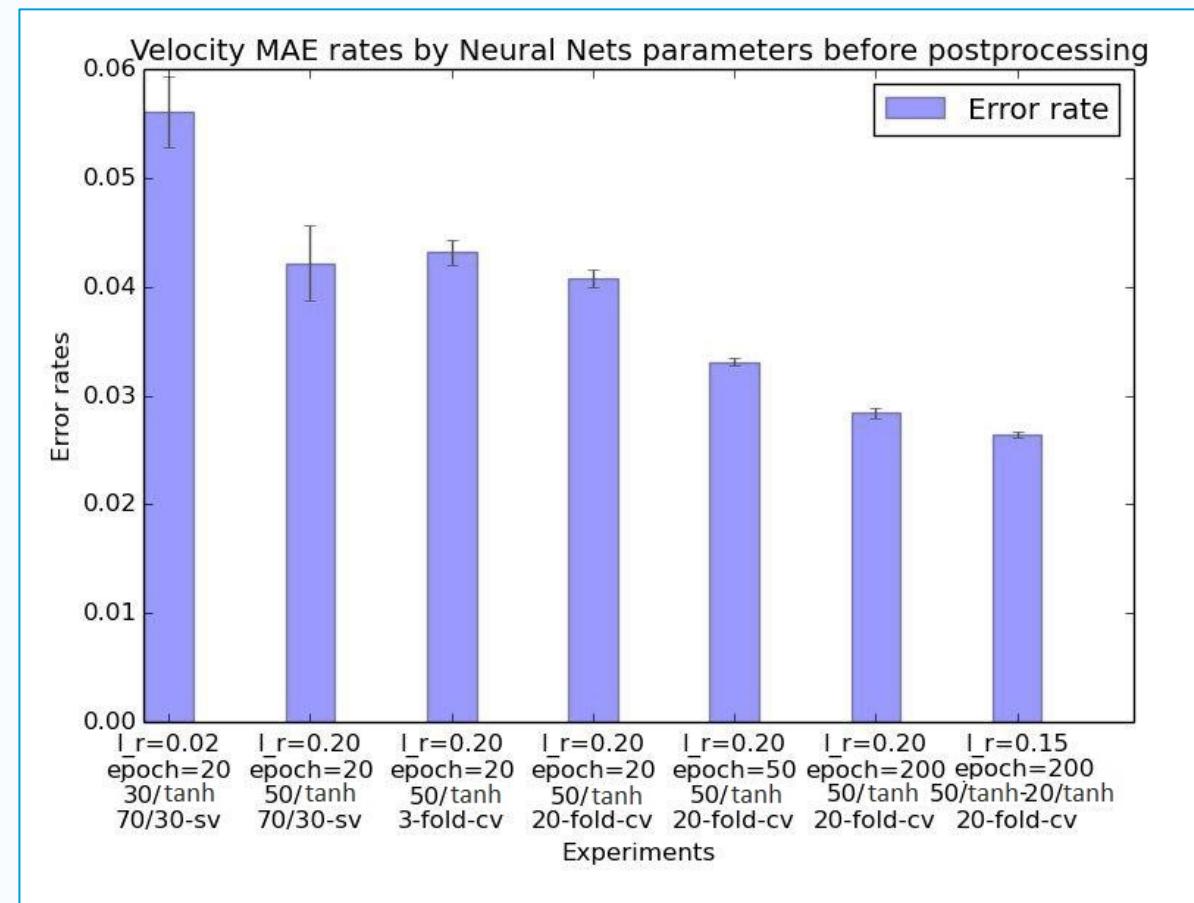
$$MAE(a, X) = \frac{1}{l} \sum_{i=1}^l |a(x_i) - y_i|$$

- тяжелее минимизировать
- штраф за сильное отклонение гораздо меньше



Эксперименты для предсказания скорости источника возмущения

- l_r – скорость обучения,
- **k -fold-cv** – кроссвалидация по k блокам,
- **70/30-sv** – обычная валидация при разбиении 70 на 30 (*simple validation*)
- **epoch** – количество эпох
- **$m/tanh$ - $n/tanh$** – m нейронов на первом скрытом слое, n нейронов на втором скрытом слое с функцией активации гиперболический тангенс

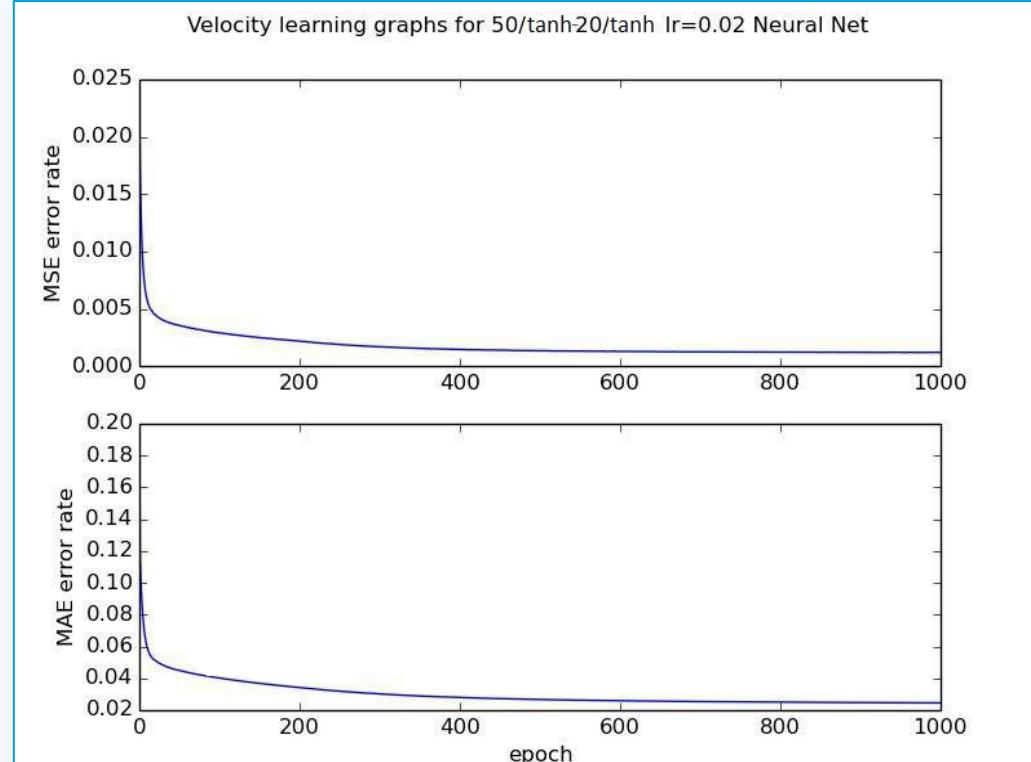
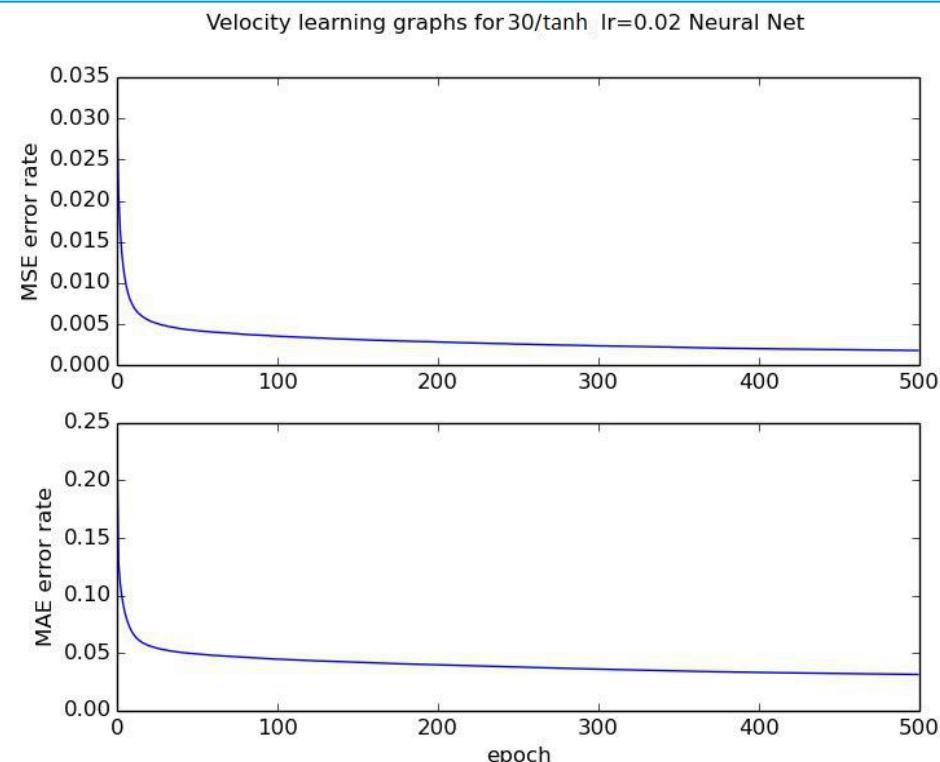


Значения среднемодульных отклонений и их дисперсий



Эксперименты для предсказания скорости источника возмущения

Графики зависимостей MSE, MAE от количества эпох



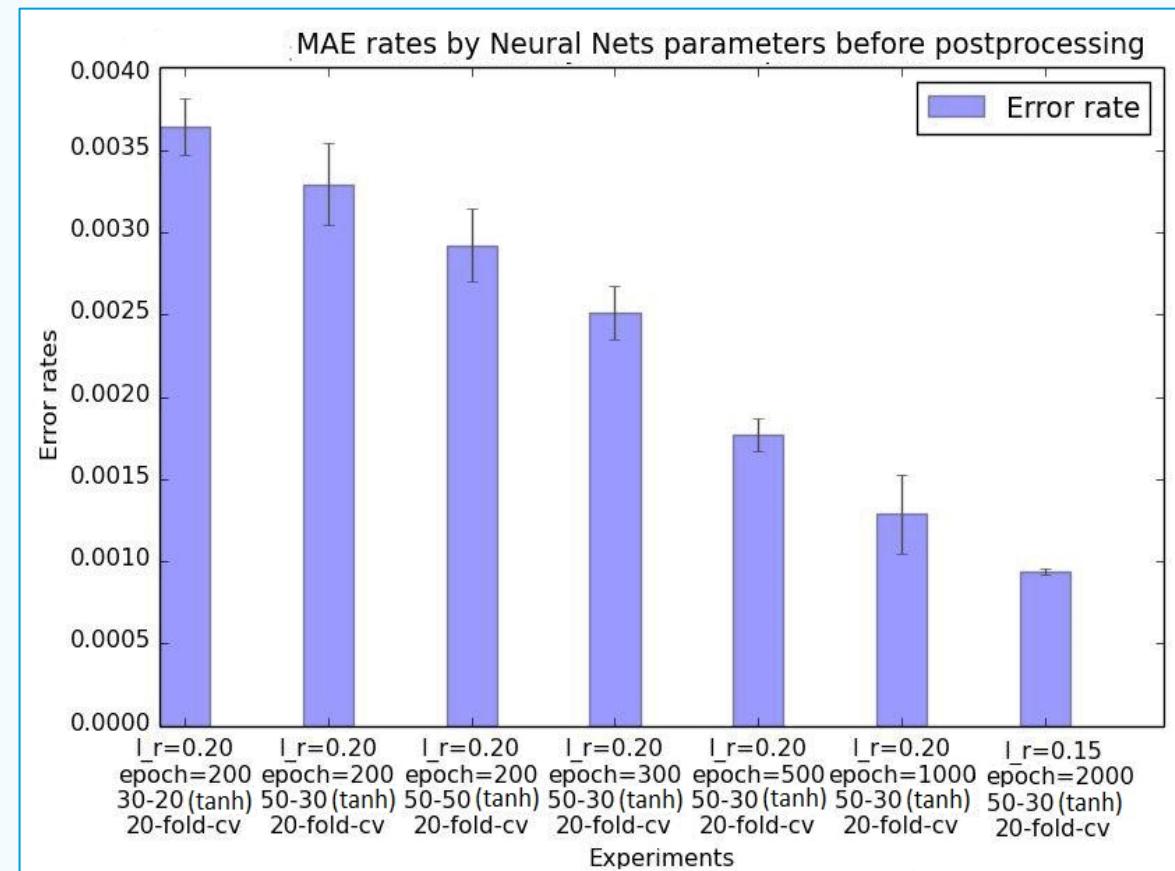
однослойная нейронная сеть, 30 нейронов
на первом слое с функцией активации
гиперболический тангенс

двухслойная нейронная сеть, 50 нейронов
на первом скрытом слое, 20 нейронов на
втором



Эксперименты для предсказания глубины источника возмущения

- l_r – скорость обучения,
- **k -fold-cv** – кроссвалидация по k блокам,
- **70/30-sv** – обычная валидация при разбиении 70 на 30 (*simple validation*)
- **epoch** – количество эпох
- **$m/tanh$ - $n/tanh$** – m нейронов на первом скрытом слое, n нейронов на втором скрытом слое с функцией активации гиперболический тангенс

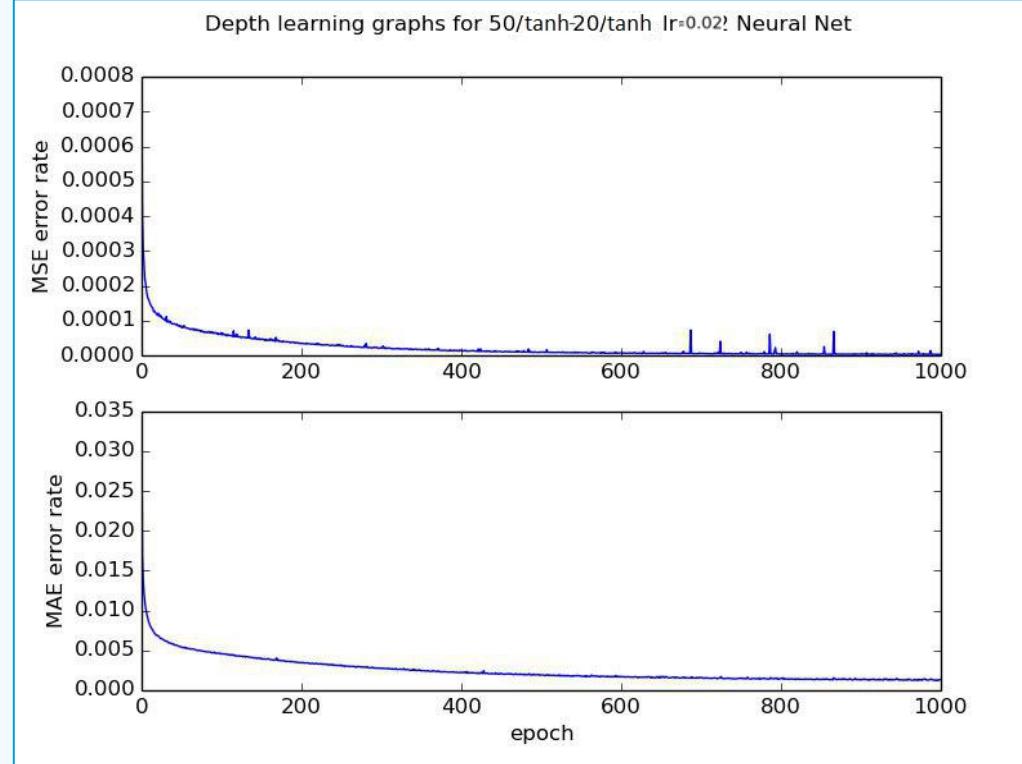
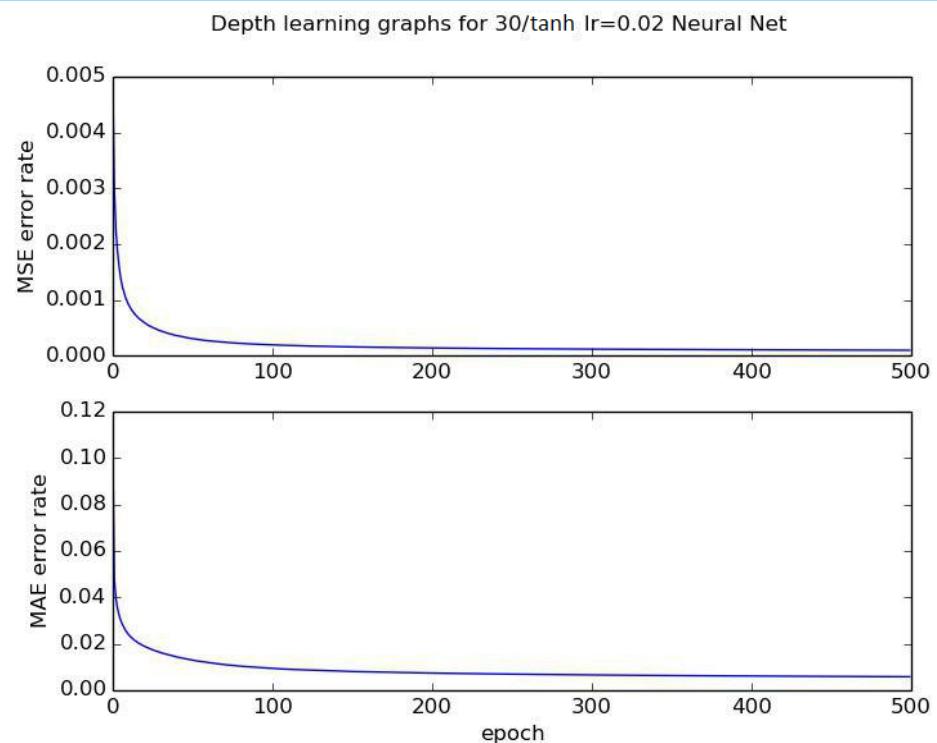


Значения среднемодульных отклонений и их дисперсий



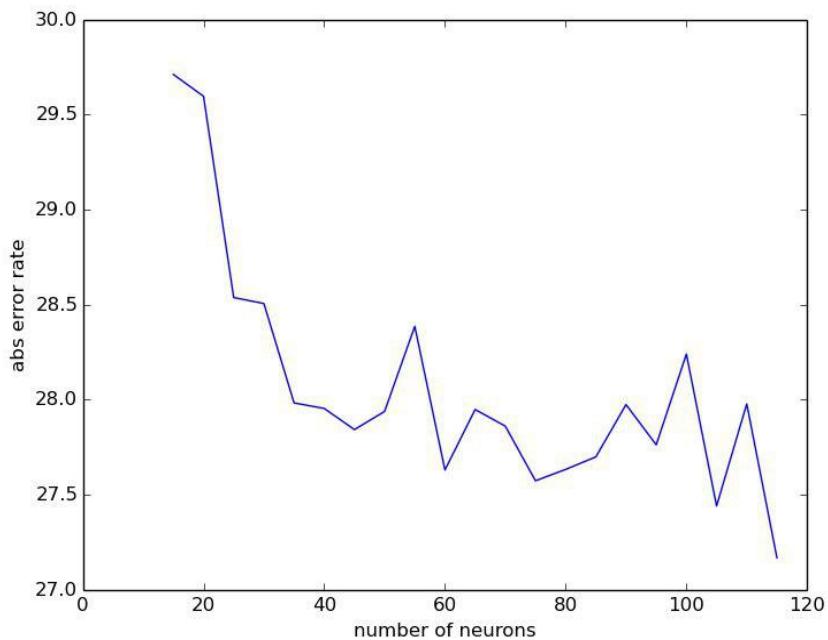
Эксперименты для предсказания глубины источника возмущения

Графики зависимостей MSE, MAE от количества эпох



однослойная нейронная сеть, 30 нейронов
на первом слое с функцией активации
гиперболический тангенс

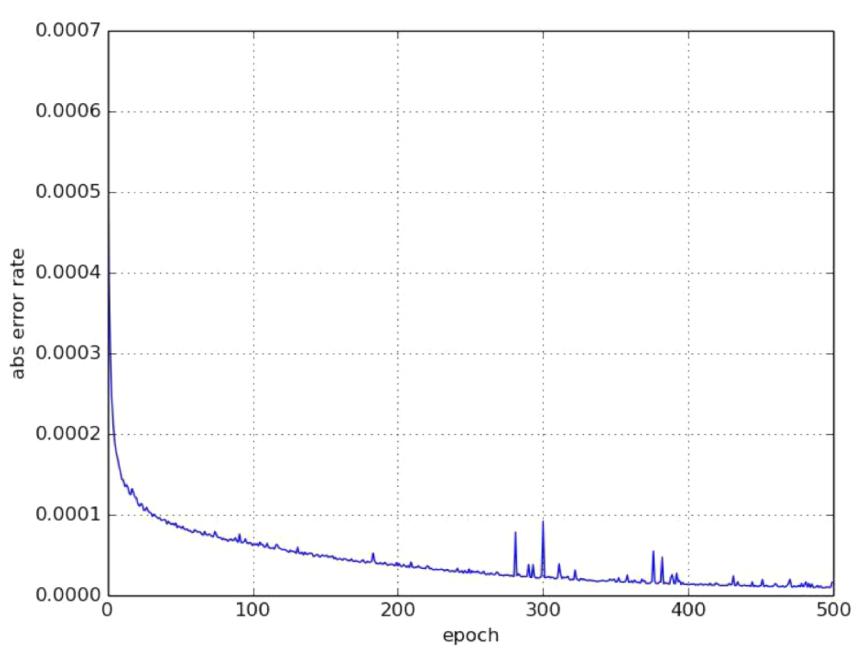
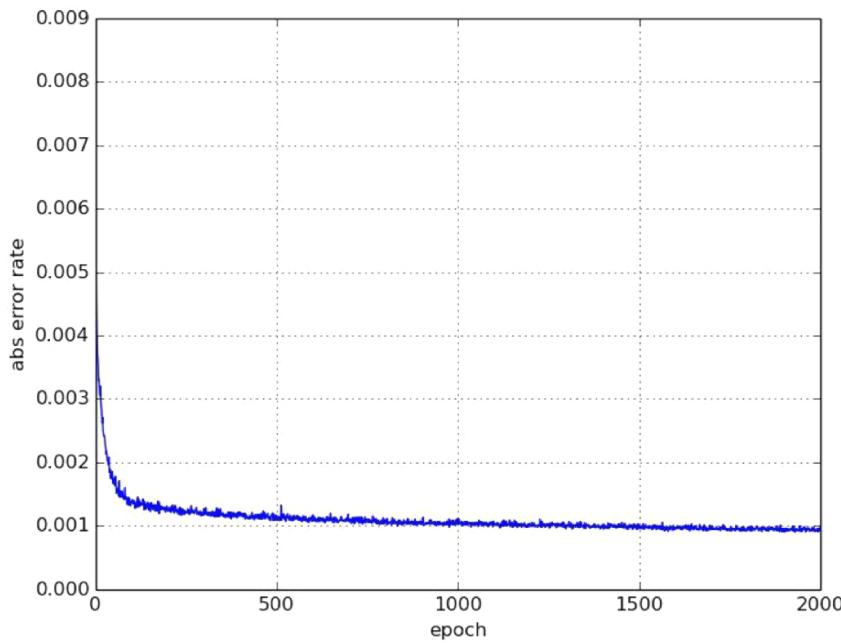
двуслойная нейронная сеть, 50 нейронов
на первом скрытом слое, 20 нейронов на
втором



Обозначения:

numbers of neurons – количество нейронов,
abs error rate - средний модуль отклонения
epoch – номер эпохи

График зависимости среднего модуля отклонения до постпроцессинга от количества нейронов в скрытом слое однослойной нейронной сети



Графики зависимости среднего модуля отклонения от количества эпох
Слева – для глубины. Справа – для скорости



Применение полученной модели нейронной сети

Обозначения:

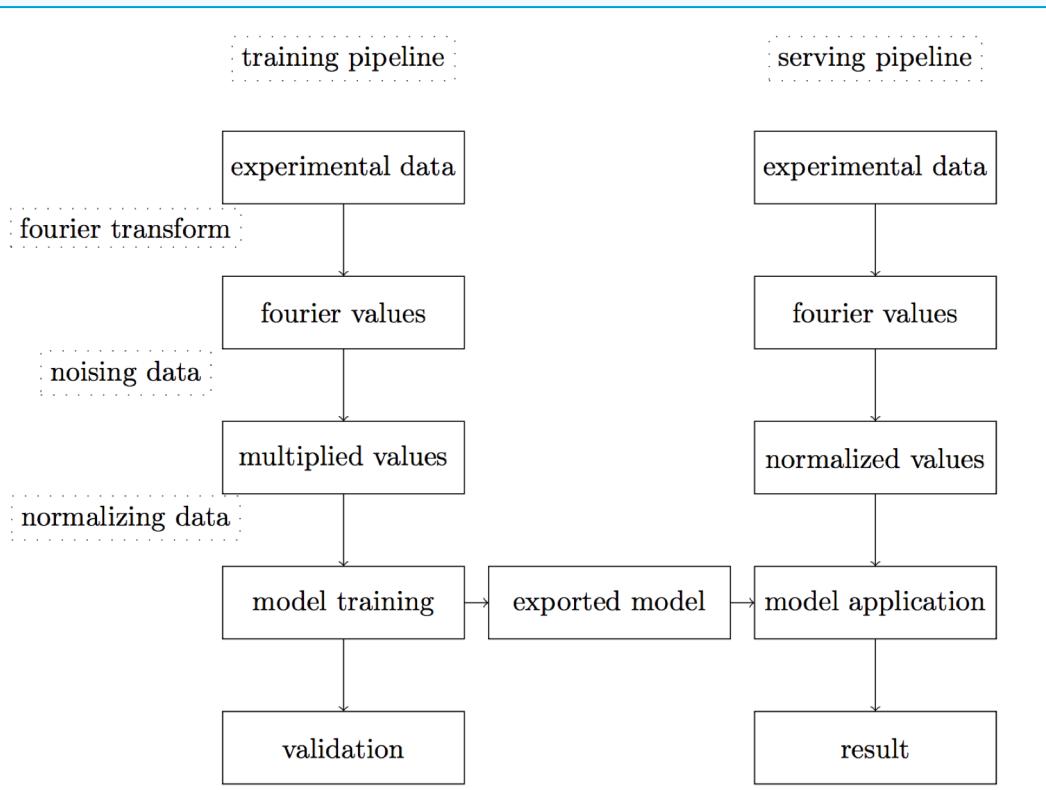
Таблица сравнения результатов с полученными в результате физического эксперимента

array_points – массив значения амплитуд
Depth – выдаваемая глубина
Velocity – выдаваемая скорость
Н – фактическая глубина
V – фактическая скорость

experiment	array_points	Depth, мм	Velocity, мм/с	Н,мм	V, мм/с
1	[2.63, 4.52, 4.19]	173.072	478.193	175	482
2	[3.6, 8.11, 10.68]	170	665.825	175	667
3	[4.93, 13.24, 15.1]	171.407	752.935	175	753
4	[2.02, 36.6, 64.68]	174.905	936.668	175	938
5	[1.88, 2.98, 2.32]	66.077	277.129	66	280
6	[2.46, 3.31, 2.7]	66.802	338.377	66	345
7	[2.69, 3.26, 4.97]	66.034	423.519	66	425
8	[0.09, 2.9, 4.18]	96.234	457.177	96	470
9	[0.25, 3.74, 6.77]	95.760	624.421	96	625
10	[1.64, 20.09, 18.38]	95.999	843.9	96	845
11	[2.24, 6.14, 6.46]	126.309	675.429	126	650
12	[4.4, 13.99, 12.6]	126.208	839.759	126	841
13	[4.79, 21.4, 38.2]	126.813	937.406	126	938
14	[1.06, 40.51, 66.83]	125.808	1055.386	126	1056
15	[1.43, 1.68, 4.46]	103.134	322.146	103	324
16	[2.09, 3.15, 2.1]	133.552	288.717	133	294
17	[2.45, 7.34, 11.87]	133.612	564.718	133	565



Разработка программного комплекса [1]



Блок-схема обучения нейронной сети

```
Во сколько раз больше получаем данных на этапе завалки?
MULTIPLY_NUMBER = 100
# Интервал значений
MULTIPLY_RANGE = (-1, 1)

TEST_FILE_NAME = None
# TEST_FILE_NAME = './data.csv'

# SPLIT_SEED = 2
NOISING_SEED = 0

TRAIN_FILE_NAME = './data.csv'

# TARGET = 'Depth'
TARGET = 'Velocity'

# data_columns - индексы колонок в файле из которых берутся данные
DATASET_SETTINGS = {'data_columns':[1, 3, 5]} # , 'addition_range':MULTIPLY_RANGE}

MODEL_SETTINGS = {
    'model_name': 'NeuralNet',
    'learning_rate': 0.1,
    'layers':[{'size':20, 'activation':'relu'}],
    # 'layers':[{'size':10, 'activation':'tanh'}, {'size':10, 'activation':'tanh'}],
    'number_of_epochs': 10
}

"pipeline.py"
0.0000e+00
Epoch 10/10
32/1201 [.....] - ETA: 0s - loss: 0.0050 - acc: 0.000
480/1201 [=====>.....] - ETA: 0s - loss: 0.0068 - acc: 0.000
992/1201 [=====>.....] - ETA: 0s - loss: 0.0066 - acc: 0.000
1201/1201 [=====] - 0s 11lus/step - loss: 0.0064 - acc: 0.0000e+00
Mean sqr error: 105297.09706796275
Mean abs error: 264.2840447703933
[0.006006679658252334, 0.06211775333520979]
agrin@agrin-Aspire-ES1-131:~/Projects/hydrodynamics$
```

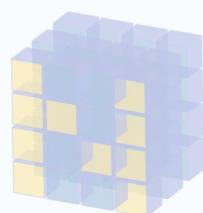
Скрин запуска скрипта pipeline.py в процессе обучения нейронной сети



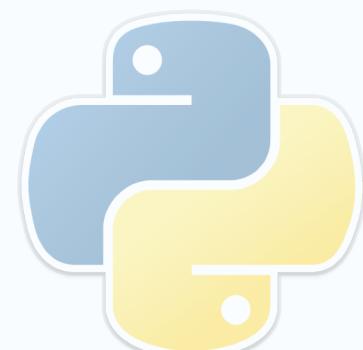
Разработка программного комплекса [2]

```
agrin@agrin-ThinkPad-Edge: ~/Projects/hydrodynamics
blurer.py      generate_response.py  journal.txt      repo          validation.py
data           graphs_builder.py    pipeline.py      report_depth.txt
agrin@agrin-ThinkPad-Edge:~/Projects/hydrodynamics$ cd ../..
agrin@agrin-ThinkPad-Edge:~/Projects/hydrodynamics$ ls
blured_report.txt  data.csv        journal_depth.txt  __pycache__      report.txt
blurer.py         generate_response.py  journal.txt     repo          validation.py
data              graphs_builder.py  pipeline.py     report_depth.txt
agrin@agrin-ThinkPad-Edge:~/Projects/hydrodynamics$ vim generate_response.py
agrin@agrin-ThinkPad-Edge:~/Projects/hydrodynamics$ python3 generate_response.py
/usr/lib/python3.4/importlib/_bootstrap.py:321: FutureWarning: Conversion of the second argument of
issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64
== np.dtype(float).type`.
    return f(*args, **kwds)
Using TensorFlow backend.
2018-06-03 15:29:03.018627: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports i
nstructions that this TensorFlow binary was not compiled to use: AVX2 FMA
./data/test_17.json
[2 15 7 21 11 22]
Depth: 133.61188769340515
Velocity: 562.7180337905884
Exception ignored: ...: second method Session.__del__ of <tensorflow.python.client.session.Session object at 0x7fdc943506a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.4/dist-packages/tensorflow/python/client/session.py", line 712, in __
del__
TypeError: 'NoneType' object is not callable
agrin@agrin-ThinkPad-Edge:~/Projects/hydrodynamics$
```

Скрин запуска программы,
Depth – результирующая глубина,
Velocity - результирующая скорость



NumPy





Научные и практические результаты работы

1. Построена оптимальная модель нейронной сети со следующей конфигурацией: 3-50/tanh-20/tanh-1 и оптимальными параметрами: *скорость обучения – 0,2; количество эпох – 200; валидационная процедура – 20-fold-cv*
2. Предложен метод стохастического градиентного спуска (SGD), являющийся эффективным для обучения полученной модели нейронной сети
3. Осуществлено сопоставление полученных параметров крылового профиля и кругового цилиндра с данными лабораторного эксперимента. Установлено совпадение результатов с достаточной точностью
4. Разработан программный комплекс, позволяющий эффективно и с минимальными затратами по времени решать задачу определения параметров источника возмущения по данным, снятым с водной поверхности
5. Установлено, что для решения обратной задачи генерации поверхностных волн применимы методы машинного обучения, в частности нейронные сети



**СПАСИБО
ЗА ВНИМАНИЕ!**