Е.А.ВОРОНИН, А.С.СЫЧУГИНА

**НЕЙРОСЕТЕВОЙ ПОДХОД К РЕШЕНИЮ ОБРАТНОЙ ЗАДАЧИ ГЕНЕРАЦИИ ПОВЕРХНОСТНЫХ ВОЛН**

В текущей статье рассмотрен нейросетевой подход для идентфиикации источника возмущения по данным, снятым с его свободной поверхности. Построена модель нейронной сети на основе данных, полученных в ходе лабораторного эксперимента, для крылового профиля и кругового цилиндра. Для построенной нейросетевой модели проведены численные эксперименты для предсказания глубины и скорости. Результаты расчёта сопоставлены с данными эксперимента. Разработан программный комплекс, позволяющий эффективно и с минимальными затратами по времени решать задачу определения параметров источника возмущения по данным, снятым с водной поверхности.

***Ключевые слова:*** *обратная задача генерации поверхностных волн; нейронные сети; обучение нейронных сетей; метод стохастического градиентного спуска; мониторинг состояния нейронной сети; спектральный анализ; алгоритм обратного распространения ошибки; задача регрессии; сети прямого распространения сигнала*; *многослойный персептрон.*

ВВЕДЕНИЕ

В настоящее время освоение Мирового океана приобретает все более возрастающее значение. Требуется создавать новые методы получения информации и обработки данных о протекающих в морской среде процессах [1]. Необходимость изучения процессов в толще морской среды по данным о свободной поверхности привела к созданию нового направления компьютерной радиотомографии морской среды, которое позволяет формировать образы явлений, наблюдаемых в приповерхностных областях среды, недоступных радиосигналам [6].

Постановка обратной задачи генерации поверхностных волн заключается в следующем: следует восстановить внутреннюю структуру потока тяжелой жидкости (или определить параметры источника возмущения) по данным, снятым с его свободной поверхности. Источник возмущения можно заменить его моделью в виде системы гидродинамических особенностей (источников, стоков, вихрей), отвечающей этому источнику в безграничном потоке [2].

Но восстановление внутренней структуры потока сопряжено с преодолением значительных трудностей. Задача продолжения полей в сторону источников относится к классу некорректно поставленных задач [7]. Решения таких задач неустойчивы к малым изменениям исходных данных, классическими методами решить практически невозможно во всех реальных случаях. Кроме того, метод регуляризации Тихонова, позволяющий находить приближенное решение некорректно поставленных задач, также наталкивается на данную трудность [7].

Впервые решение обратной задачи было основано на технике аппроксимаций Паде. Однако предложенная им процедура применима лишь в случае абсолютно точного задания исходных данных [8,9]. Способ решения, основанный на замене обтекания потоком неоднородности эквивалентной системой гидродинамических особенностей, предложен в работе [1].

В данной работе приведен новый подход к решению обратной задачи генерации поверхностных волн – применение методов машинного обучения, а именно теории нейронных сетей, который широко применяется в задачах обработки данных и их анализа [Хайкин, Островский]. Для прикладной реализации полученных результатов был разработан программный комплекс, с помощью которого по экспериментальным данным, содержащим результаты измерений наклона свободной поверхности, можно определить параметры источника возмущения – глубину и скорость.

**Цель исследования и решаемые задачи**

Разработать программный комплекс, предназначенный для определения параметров источника возмущений по данным о вызываемых им поверхностных волнах. Цель достигается путем решения следующих задач:

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

**Формирование обучающей выборки**

Пусть крыловой профиль (круговой цилиндр) равномерно движется по горизонтали со скоростью V на достаточно большой глубине в жидкости со свободной границей. В системе координат, связанной с крыловым профилем (круговым цилиндром), задача сводится к изучению обтекания крылового профиля (кругового цилиндра) равномерным потоком со свободной границей.

Пусть в качестве гидродинамической особенности возьмем точечный вихрь. Тогда общее выражение для профиля свободной границы можно представить в следующем виде (формула Келдыша) [1, 2]:



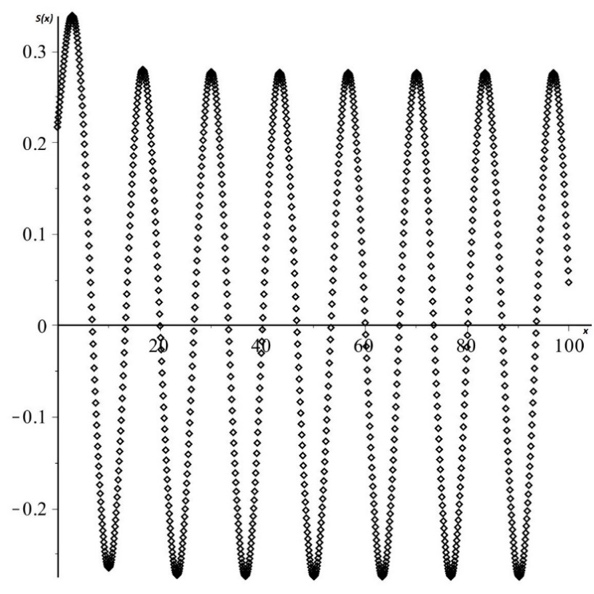
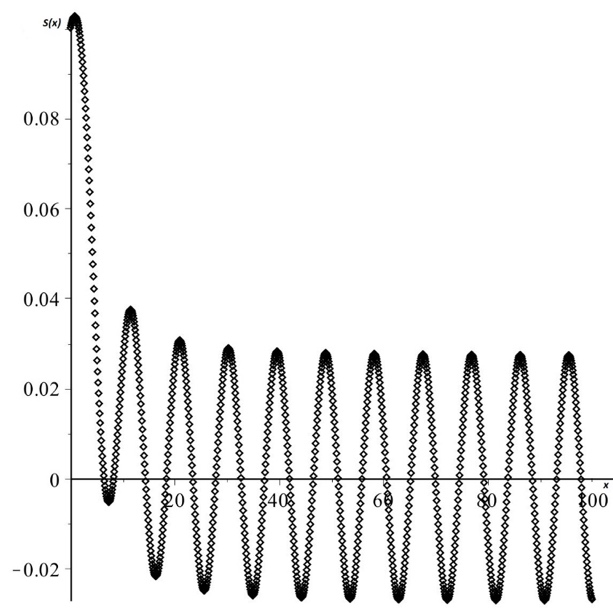
Пусть точечный вихрь интенсивности Г локализован в точке , , , тогда:

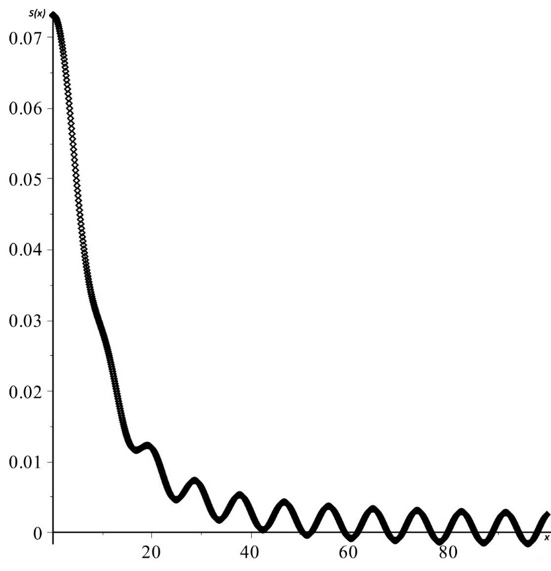
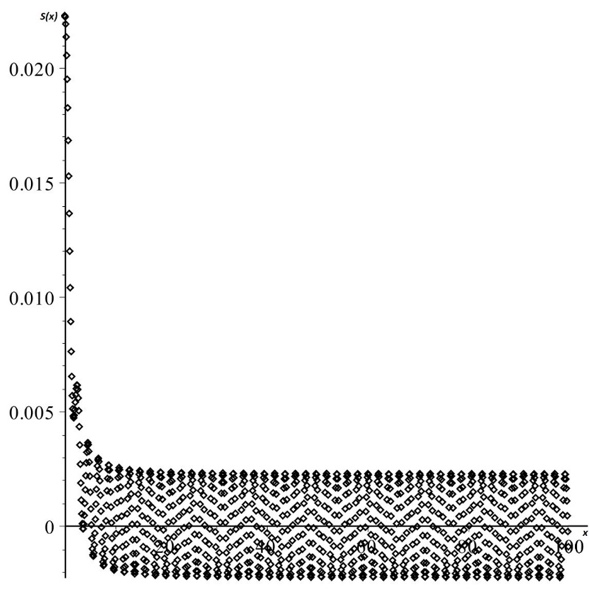


В работе [1] была получена следующая задача Коши для точечного вихря интенсивности Г, локализованного в точке **:**



Численно решая данную задачу методом Рунге-Кутты 4-го порядка (рис , можно получить обучающую выборку. Но для того, чтобы учесть окружающие возмущения, в частности вязкость жидкости, приблизиться к реальным условиям окружающей среды, выгоднее использовать данные для обучения, полученные из реального эксперимента.

1)**2)

3)4)

***Рис.*** *Решение задачи, соответствующей обтеканию точечного вихря интенсивности , локализованного в точке , при заданных параметрах: 1) , , ,  2): , , ,  3) , , , *

*4) , , , *

**Получение данных из лабораторного эксперимента**

В Институте проблем механики РАН было создано специальное оборудование и разработан метод отражённой сетки, позволяющий наблюдать возмущение свободной поверхности жидкости с высокой точностью. Проводились измерения углов наклона возмущённой поверхности воды при движении в её толще модели цилиндра и крылового профиля и были получены значения тангенса угла наклона поверхности воды в точках с горизонтальными координатами при различных скоростях V движения цилиндра (крылового профиля) [1]

**Постановка задачи в контексте нейронных сетей**

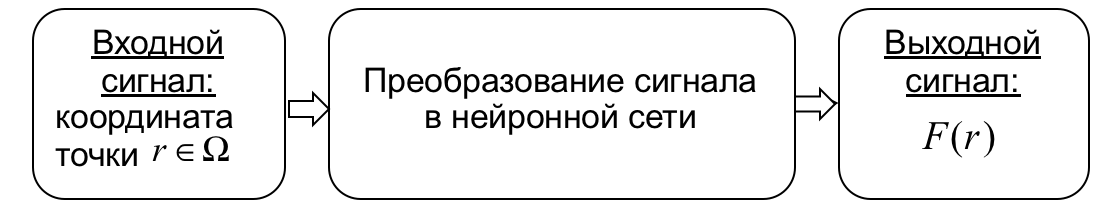
Формально задача обучения нейросети формулируется как задача аппроксимации, постановка которой имеет следующий вид.

Пусть задана область  и заданы два множества точек:

 – обучающее множество;

 – целевое множество.

 – значения функции в точках обучающего множества,  – аппроксимирующая функция, которая строится с помощью нейронной сети как непрерывное отображение . Необходимо построить нейронную сеть (аппроксимирующую функцию), которая будет принимать совпадающие значения (с заданной точностью) не только на данных участвовавших в обучении, но и на данных контрольного множества (целевого) не участвовавших в обучении. На вход поступают координаты точки  , далее происходит преобразование сигнала в нейронной сети. На выходе получаем выходной сигнал  . В таких задачах всегда используется обучение с учителем, то есть для каждого обучающего входного вектора имеется обучающий выходной вектор. Тем самым логично использовать сети прямого распространения сигнала такие как многослойный персептрон, обучаемый алгоритмами, основанными на методе обратного распространения ошибки [Хайкин островский ].



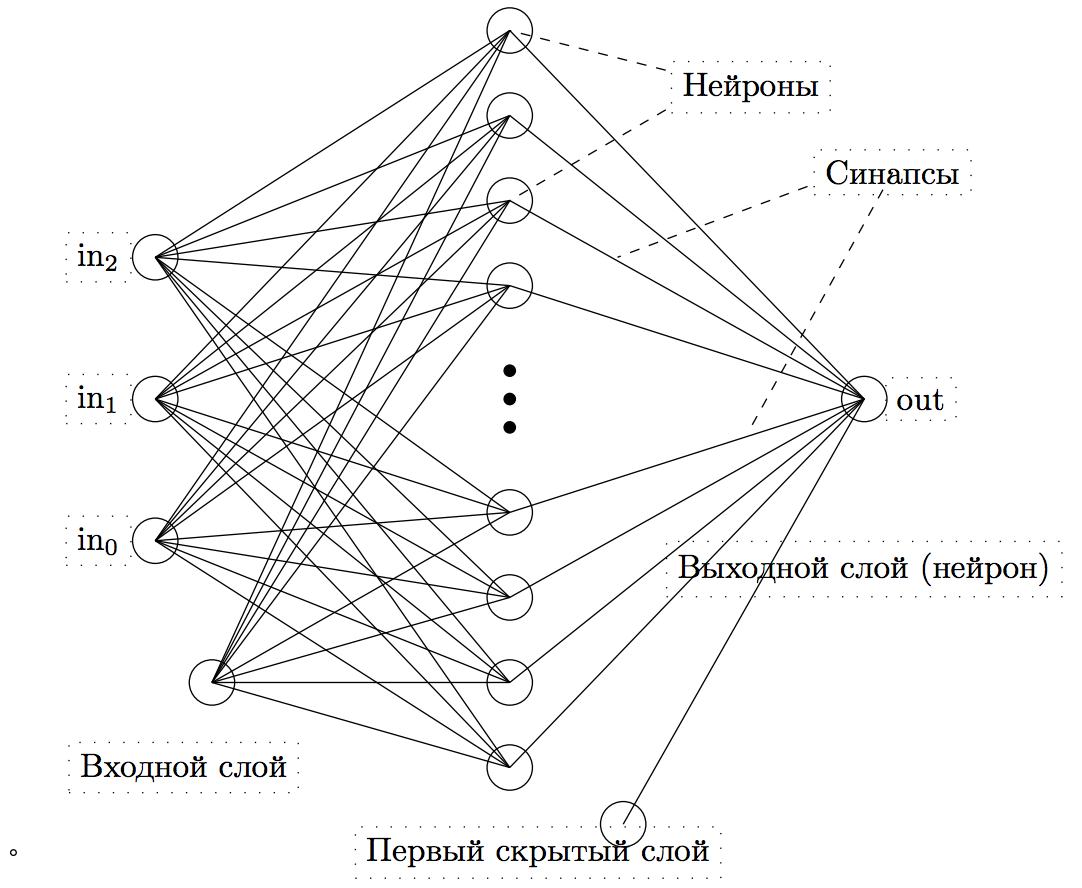
***Рис.*** *Схема преобразования сигнала в нейронной сети*

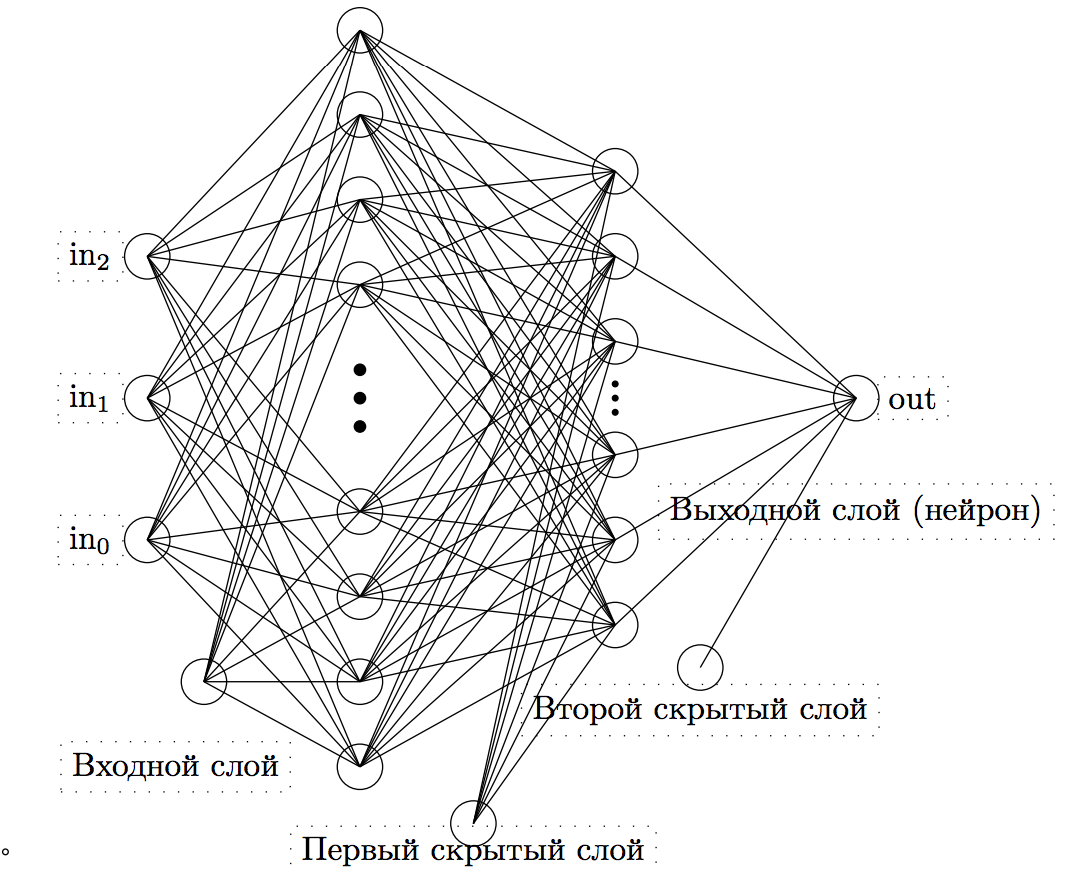
**Выбор архитектуры нейронной сети для идентификации источника возмущения**

Важной проблемой наряду с получением данных для обучения и их обработкой служит нахождение оптимальной архитектуры нейронной сети, а именно поиском количества слоем, числа нейронов на скрытом слое.

После проведения достаточного количества численных экспериментов была выбрана архитектура полносвязной (в паре соседних слоев все нейроны связаны между собой) нейронной сети, которая имеет следующий вид ориентированного графа (рис. 1). Во входном слое имеется 3 нейрона (вершины), на которые в процессе обучения и применения приходят значения амплитуд. В первом (скрытом) слое имеется некоторое количество (от 20 до 70 в зависимости от эксперимента) нейронов с активационной функцией , либо фукцией активации гиперболический тангенс, каждый из которых соединен синапсом (ребром) с каждым из входящего слоя. Такую сеть здесь и далее будем обозначать следующим образом: 3-20-1. Это означает, что сеть имеет 3 входных, 20 скрытых и 1 выходной нейрон.

В некоторых экспериментах добавляется второй слой, каждый нейрон из которого в свою очередь соединен с каждым нейроном из первого слоя (рис. 2). Каждый нейрон из последнего из промежуточных слоев (первого в случае однослойной нейросети и второго – в случае двухслойной) соединяется с единственным нейроном выходного слоя, отвечающего за результат применения нейронной сети, а именно – глубину, либо скорость.



*Рис 1. Архитектура полносвязной нейронной сеть с одним скрытым слоем, используемой для обучения* 

*Рис. 2 Архитектура полносвязной нейронной сеть с двумя скрытыми слоями, используемой для обучения*

*Задание настроек обучения нейронной сети*

1. MODEL\_SETTINGS = {
2. 'model\_name':'NeuralNet',
3. 'learning\_rate':0.20,
4. 'layers':[{'size':50, 'activation':'relu'}, {'size':20, 'activation':'relu'}],
5. # 'layers':[{'size':10, 'activation':'tanh'}, {'size':10, 'activation':'tanh'}],
6. 'number\_of\_epochs':200
7. }

ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

***Применение спектрального анализа***

Для выявления характерных признаков процесса и сокращения обучающей выборки было принято решение к данным, полученным в ходе лабораторного эксперимента, применить спектральный анализ. Спектр инвариантен относительно смещения результатов наблюдения. Визуально полученный спектр Фурье представлен на рис. . Как видим, частоты имеют примерно одинаковые значения и можно выявить три характерные частоты. Значения частот были отброшены в виду их низкой вариативности и слабой связи между целевыми метками – H и V (глубиной и скоростью соответственно). Из всего этого следует решение о том, что следует сгенерировать выборку для обучения с тремя входами – значениями получившихся амплитуд 

***Зашумление полученной выборки***

Считается, что нейронная сеть обладает хорошими свойствами обобщения, если в результате обучения она порождает правильный вывод для большинства вводимых образцов из набора тестовых данных.

Несмотря на то, что для применения нейронных сетей обычно во всем используют экспериментальный подход, для определения размера обучающей выборки, как правило, следует придерживаться следующего неравенства [10, 29]:



где – число учебных образцов, – число весовых коэффициентов в сети,  – доля ошибок, которая допустима в ходе тестирования.

Для нашей архитектуры нейронной сети имеем: число весовых коэффициентов – , доля ошибок, которая допустима в ходе тестирования – .

Получаем, что для заданной ошибки количество примеров обучения должно в 10 раз превосходить количество весовых коэффициентов сети. Тем самым, для хорошего обобщения это совершенно недостаточный объем выборки для обучения, поэтому было решено на основе имеющихся данных сгенерировать выборку большего размера путем зашумления данных. Кроме того, это поможет нам избежать излишне точного соответствия нейронной сети конкретному набору обучающих примеров или переобучения.

Тем самым, была сформировано выборка большего объема путем 5% зашумления (выбор интервала зашумления был выявлен путем эксперимента). Кроме того, это помогло там частично достичь хорошего обобщения. Свойство обобщения показывает то, насколько хорошо сеть выполняет свои функции на новых данных, которые не участвовали в процессе обучения.

А также такое решение помогло избежать переобучения (излишне точного соответствия нейронной сети конкретному набору обучающих примеров).

***Нормирование данных***

В различных типах выборок диапазоны различных признаков могут существенно различаться. Но как видно из архитектуры нейронной сети все они подаются на один и тот же сумматор. Для того, чтобы провести процесс обучения по вариативности факторов, вначале было проведено нормирование данных. Для этого мы привели к относительным единицам в интервале (0,1) путем деления каждого параметра на максимальное значение.

**Метод стохастического градиентного спуска для обучения нейронной сети**

обучение нейронной сети сводится к минимизации функции ошибки, путем корректировки весовых коэффициентов синаптических связей между нейронами.

Выражение для среднеквадратичной ошибки можно представить в следующем виде:



Здесь yi – фактический результат, di - желаемый отклик сети.

Таким образом, методика использования градиента в алгоритме обучения нейронной сети понятна. Здесь стоит ключевой вопрос: «В каком направлении и на какую величину менять веса, чтобы достичь   
минимума функции ошибки за минимальное число шагов?». Очевидно, что в том направлении, в котором функция ошибки уменьшается быстрее всего.   
Именно это направление и указывает градиент (точнее антиградиент).

Пусть у нас есть некий функционал:



Вычислим градиент от него (вектор частных производных):



После подстановки и преобразований имеем:



Обновление весов осуществляется по следующей формуле, которое представляет собой некое аддитивное выражение (сумму по всем объектам выборки):



В этом и заключается недостаток данного метода. А вот при SGD мы выбираем именно один случайный объект из обучающей выборки и далее вычисляем градиент функционала только на этом объекте.   
Кроме того, преимущества метода SGD заключаются в следующем:

1. помогает избежать «застревания» модели в локальном минимуме;
2. проще в вычислениях. Помогает производить эффективные вычисления и не хранить все вычисления в оперативной памяти;
3. но не позволяет доказать существование единственности решения

Кроме того, хотелось бы отметить, что очень важно при выборке объекта случайным образом осуществлять перемешивание (shuffling), то есть случайно выбираем объекты, но попеременно из разных классов. Такой прием окажет существенное влияние на улучшение сходимости полученной модель нейронной сети.

АЛГОРИТМ???

**Мониторинг состояния построенной модели нейронной сети**

Далее после выбора архитектуры нейросети следует подумать об оптимальном выборе гиперпараметров нейронной сети, о том как модель применять модель на практике, как хорошо алгоритм будет работать на новых данных. Для этого следует применить процедуру валидации, чтобы понять, двигаемся ли мы в правильном направлении и оценить качество построенной модели нейронной сети.

Схема оценивания качества обучения состояла из нескольких этапов:

1. настройка всех алгоритмов на обучении.
2. сравнение качества на валидационной выборке
3. выбор лучшего алгоритма, того, который допускает наименьшую ошибку.
4. после того, как алгоритм выбран, мы измерим качество на контрольной выборке и проверим модель на адекватность.

Сначала использовалась простая валидационная процедура (simple validation) с простым разбиением генеральной выборки в соотношении 70 к 30 (70-тренировочное множество, размер обучающей выборки, 30-валидационное, измеряем качество на ней).

Но после некоторого количества проведенных экспериментов обнаружилось, что текущая валидационная процедура дает очень высокий разброс оценки качества предсказания алгоритма при фиксированных настройках.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **5** | 0,2 | 20 | 50-tanh | s-val-70/30 | 0,002858 | 0,0419 |
| **6** | 0,2 | 20 | 50-tanh | s-val-70/30 | 0,004266 | 0,0521 |
| **7** | 0,2 | 20 | 50-tanh | s-val-70/30 | 0,002965 | 0,0402 |
| **8** | 0,2 | 20 | 50-tanh | s-val-70/30 | 0,003948 | 0,0483 |

Для того, чтобы справиться с этой проблемой было принято решение перейти к процедуре кроссвалидации по k блокам (k-fold cross-validation) как более объективной.

Основная идея состоит в том, чтобы случайным образом разбивает данные на k непересекающихся блоков примерно одинакового размера.

Далее каждый блок по очереди будет выступать в качестве тестового. Возьмем первый блок в качестве тестового, а все остальные в качестве обучения. Обучим алгоритм на обучающей выборке, измерим качество на тестовом блоке, запомним. Возьмем второй блок в качестве тестового. Все остальные сольем в обучающую выборку, обучим все, измерим качество и т.д. Каждый блок побывает один раз тестовым. Получим k оценок качества, усредним их и получим итоговую оценку модели.



Рис. Схема работы процедуры кроссвалидации (k-fold-cross-validation)

В работе сделан вывод после проведения экспериментов с разными значениями k, что оптимальным k является 20.

**Варьирование гиперпараметров нейронной сети**

В процессе обучения проводилось варьирование гиперпараметров, что является очень важным и напрямую влияет на сходимость нейросети.

*Гиперпараметры* — значения, которые нужно подбирать вручную и зачастую методом проб и ошибок при обучении. В качестве примеров гиперапараметров можно привести следующие:

1. Момент и скорость обучения
2. Количество скрытых слоев
3. Количество нейронов в каждом слое
4. Количество эпох

После проведения численных экспериментов были найдены оптимальные настройки гиперпараметров для построенной модели:

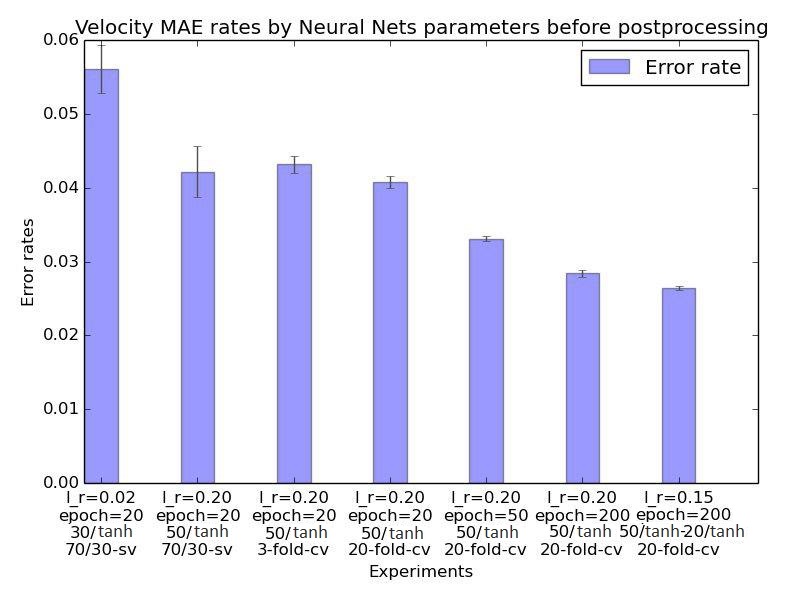
1. оптимальная скорость обучения (learning\_rate) – 0,2;
2. оптимальная конфигурация нейронной сети: 3-50/tanh-20/tanh-1
3. количество эпох – >= 200. Эксперименты показали, что, по крайней мере, до 2000 эпохи метрика качества улучшается и достаточно быстро. С большой долей уверенности можно предположить, что улучшение качества предсказания будет продолжаться при дальнейшем увеличении количества итераций обучения. Однако эксперименты были остановлены ввиду высоких и увеличивающихся пропорционально количеству эпох затрат по времени.
4. k=20 (кроссвалидация по 20 блокам)

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

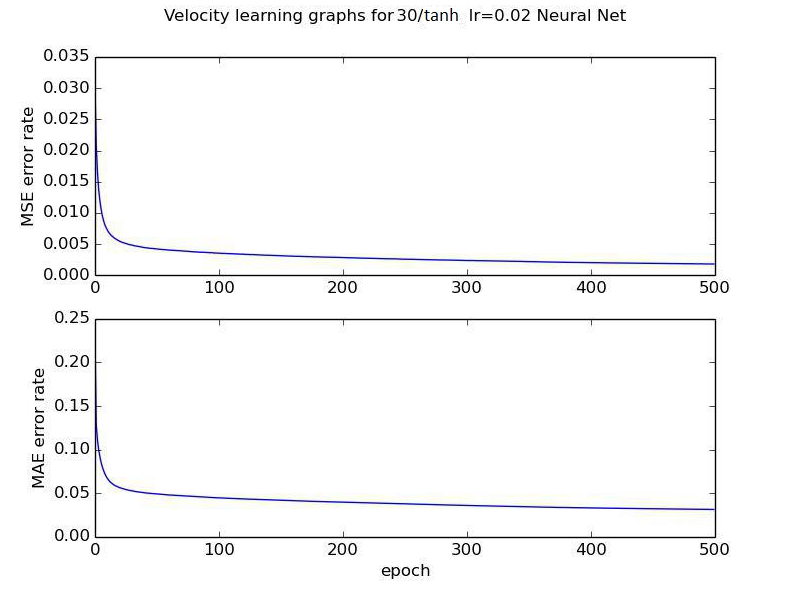
**Результаты экспериментов для предсказания скорости**

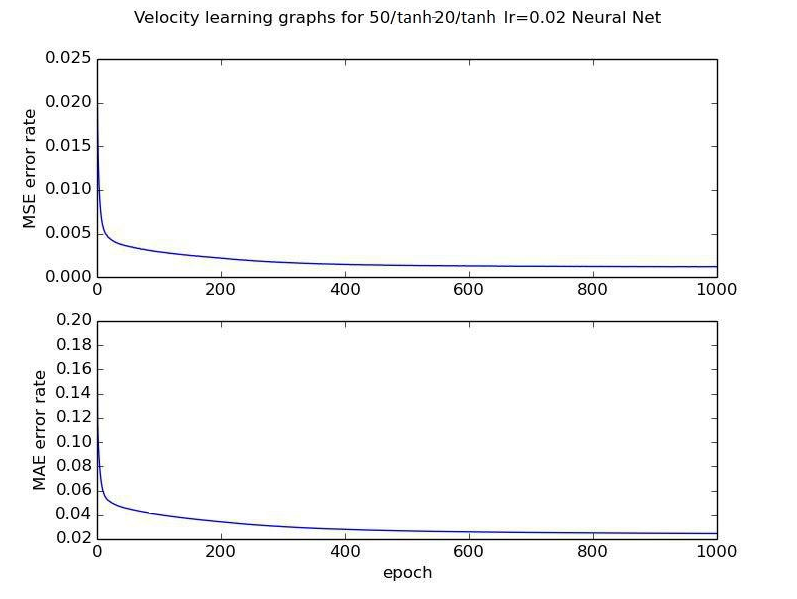
По некоторому количеству характерных экспериментов был построена гистограмма, показывающая зависимость среднего отклонения от настроек нейронной сети, а также приблизительные значения дисперсии этого отклонения.

При исходной валидационной процедуре с простым разбиением генеральной выборки в соотношении 70/30 значение дисперсии достаточно велико, при переходе к кроссвалидации по 20 блокам () наблюдаем постепенное снижение ошибки до приемлемых значений. Кроме того, наблюдается постепенное улучшение качества предсказания при увеличении количества нейронов в скрытом слое однослойной нейронной сети (рис. 12) и переходе от однослойной к двухслойной (рис. 13).



На рис. 15 представлены графики зависимости MSE и MAE от количества эпох. Как видим, значения монотонно уменьшаются с ростом количества эпох.

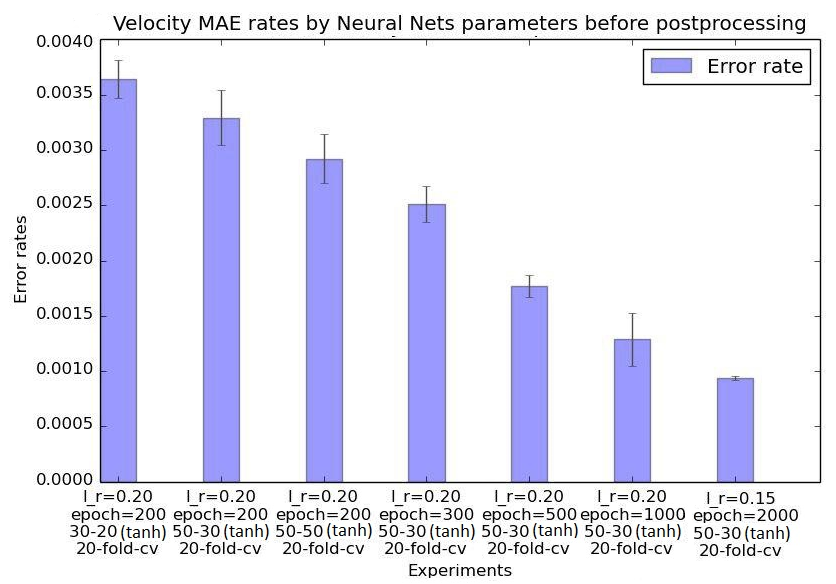
**

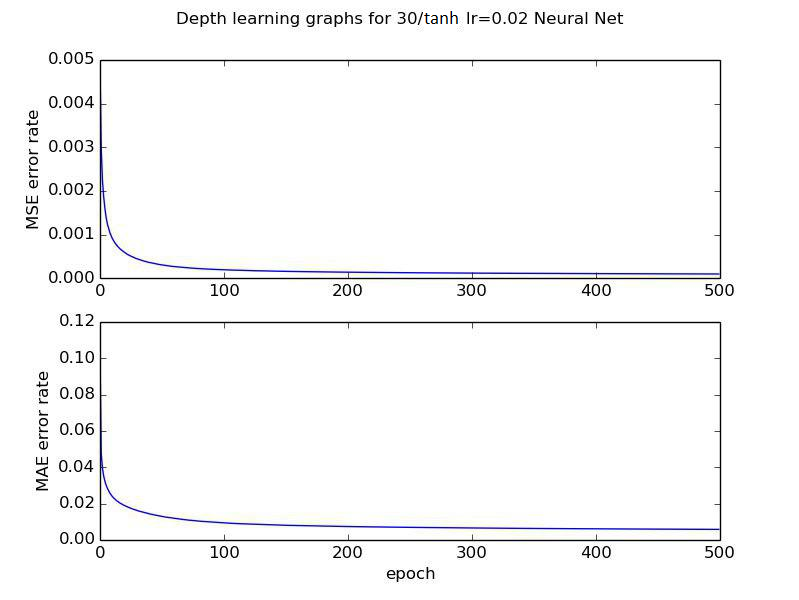
**

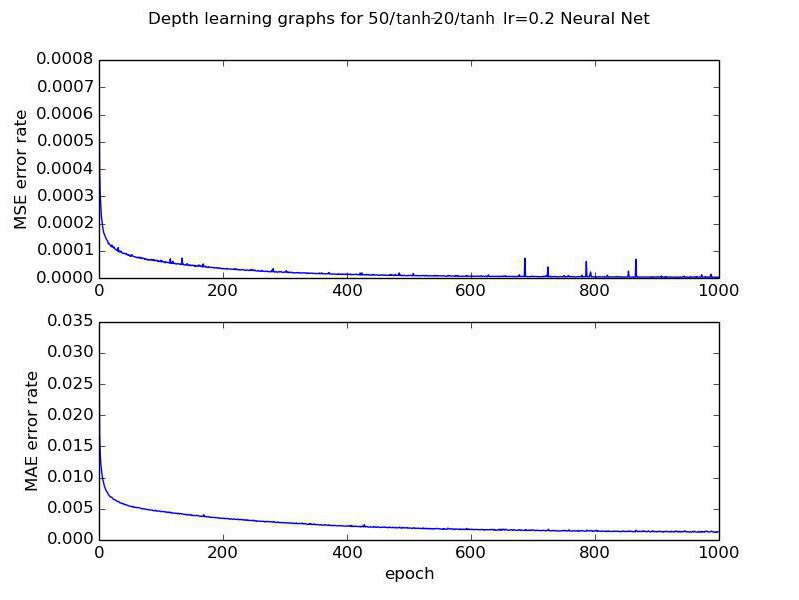
**Результаты экспериментов для предсказания глубины**

По некоторым экспериментам была построена аналогичная изображенной на рис. 14 гистограмма, показывающая зависимость среднего отклонения значения глубины  от настроек нейронной сети, а также приблизительные значения дисперсии этого отклонения.

На первых трех измерениях проиллюстрированы эксперименты с архитектурой нейронной сети при фиксированном количестве эпох () и скорости обучения (). Дальнейшие измерения показывают вычислительно сложные эксперименты с выбранной архитектурой (рис. 13) и увеличением количества эпох (). Из полученной гистограммы видим постепенное снижение ошибки при экспериментах с архитектурой сети и более стремительное (но при этом вычислительно затратное) при увеличении количества эпох.



**

******

Сравнение ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ДЛЯ ИСТОЧНИКА ВОЗМУЩЕНИЯ с наблюдаемыми в ХОДЕ ЛАБОРАТОРНОГО эксперименте

Наблюдается точное совпадение качества предсказания модели обученной нейросети с результатами, полученными в ходе лабораторного эксперимента на данных, участвовавших в процессе обучения. Для корректного сравнения некоторых значения были исключены из обучающего множества (1-3). Для них также получаем результаты с достаточной точностью.

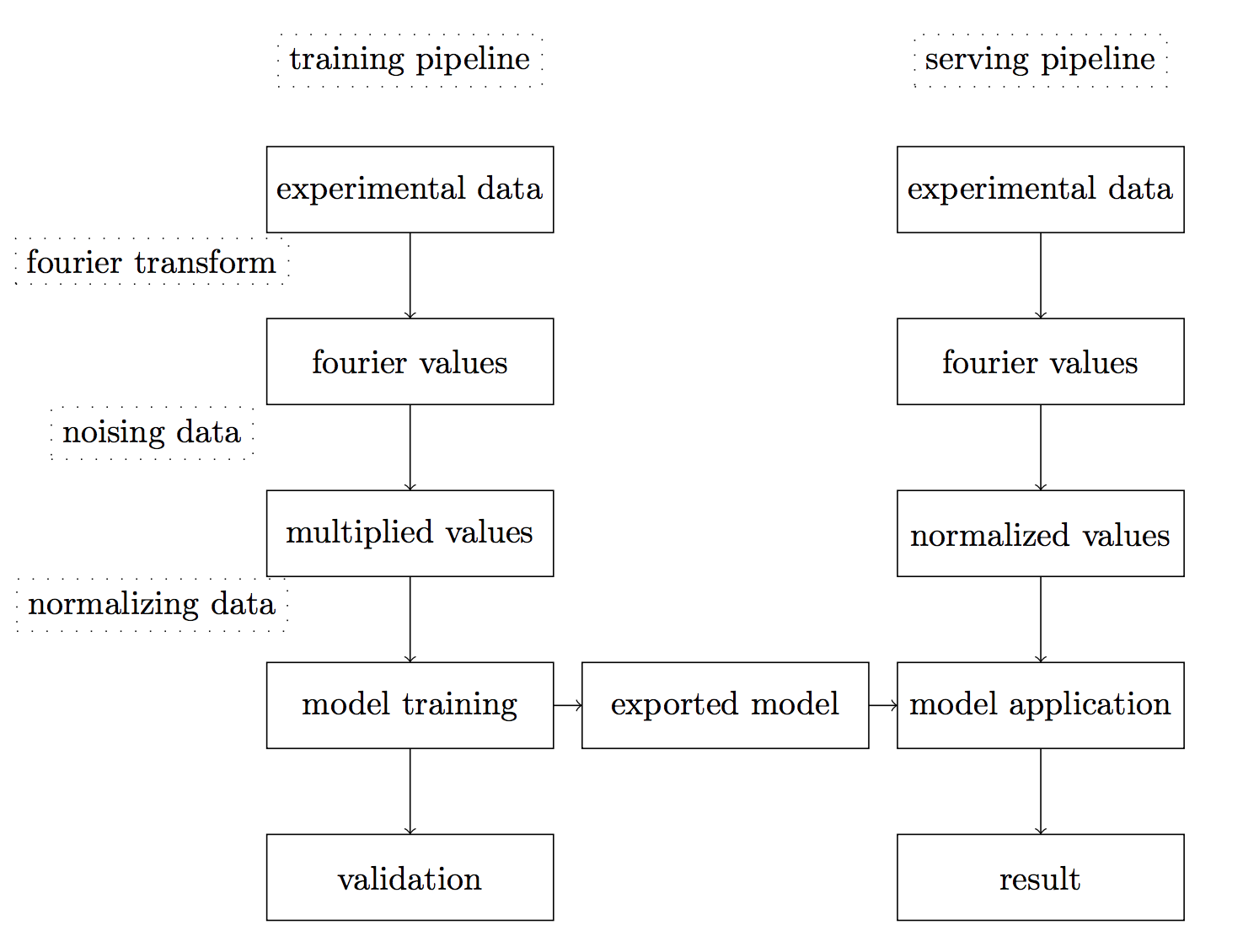
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| experiment | array\_points | Depth, мм | Velocity, мм/с | H,мм | V, мм/c |
| 1 | [2.63, 4.52, 4.19] | 173.072 | 478.193 | 175 | 482 |
| 2 | [3.6, 8.11, 10.68] | 170 | 665.825 | 175 | 667 |
| 3 | [4.93, 13.24, 15.1] | 171.407 | 752.935 | 175 | 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 |

Таблица. Результаты работы построенной нейросетевой модели

**Описание программного комплекса**

В блок-схеме (рис. 3) приняты следующие обозначения:

* *experimental data* (экспериментальные данные) – исходные данные, состоящие из значений тангенса угла наклона поверхности воды в точках с горизонтальными координатами,
* *fourier values* (значения, получившиеся после применения спектрального анализа) – несколько младших значений преобразования Фурье, примененного к исходным данным,
* *multiplied values* (размноженные значения) – увеличенный с помощью зашумления датасет,
* *model training* (обучение модели) – процесс обучения модели на значениях, получившихся после нормирования зашумленного датасета,
* *validation* (валидация) – процесс оценки качества алгоритма с помощью одной из валидационных процедур,
* exported model (экспортированная модель) – обученная модель сохраненная на диск для дальнейшего использования,
* model application (применение модели) – загрузка с диска и применение экспортированной модели,
* result (результат) – результат применения модели к новым данным (предсказание).



***Рис.*** *Структурная схема разработанного алгоритма обучения нейронной сети для предсказывания глубины и скорости*

Структура блок-схемы состоит из двух параллельных последовательностей применений цепочки преобразований входящих данных. Одна последовательность (trainig pipeline, обучающая цепь) применяется на этапе обучения модели, ее валидации и экспорта (сохранения на диск).

Вторая (serving pipeline, обслуживающая цепь) применяется на этапе использования экспортированной модели. Через первую, обучающую цепь, проходят данные всей обучающей выборки. Результатом работы этой цепи является либо оценка качества работы алгоритма машинного обучения той или иной заранее заданной валидационной процедурой, либо сохраненная на диск экспортированная модель. Вторая, обслуживающая цепь, используется для получения предсказаний на новых, неизвестных на этапе обучения, данных *–* как правило, в условиях применения обученной модели в продукте. Значительная часть описанных выше цепей совпадает *–* как по смыслу, так и на уровне программного кода. Часть, связанная с предварительной обработкой данных (от исходных до подающихся на вход алгоритму машинного обучения) совпадает буквально.

Этапу тренировки модели (с последующей валидацией либо экспортом на диск) в обучающей цепи соответствует этап применения модели в обслуживающей цепи. Результатом в этом случае является предсказание модели для входящего в цепь объекта.

**ВЫВОДЫ**

В данной статье предложен нейросетевой подход к решению обратной задачи генерации поверхностных волн. Построена оптимальная модель нейронной сети. Предложен метод стохастического градиентного спуска (SGD), являющийся эффективным для обучения полученной модели нейронной сети. Осуществлено сопоставление полученных параметров крылового профиля и кругового цилиндра с данными лабораторного эксперимента. Установлено совпадение результатов с достаточной точностью. Разработанный программный комплекс позволяет эффективно и с минимальными затратами по времени решать задачу определения параметров источника возмущения по данным, снятым с водной поверхности.

Тем самым, методы машинного обучения применимы к решению обратных задач и описанные подходы могут быть распространены на другой класс некорректно поставленных задач.

**СПИСОК ЛИТЕРАТУРЫ**

1. *Савин А.С.* Прямая и обратная задачи обтекания для плоских потоков со свободной границей // Методы, процедуры и средства аэрокосмической компьютерной радиотомографии приповерхностных областей Земли. М.: Научный мир, 1996. С. 77 – 89.
2. *Савин А.С.* Определение параметров гидродинамических особенностей в плоском потоке по данным о его свободной поверхности.// Изв. РАН.МЖГ.2001. №2. С.139–146.
3. *Каллан Р.* Основные концепции нейронных сетей.:Пер. с англ. М.: Издательский дом «Вильямс», 2001.
4. *Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д.Рудинского. – М.:Финансы и статистика, 2002. 344 с.:ил*
5. *Хайкин, С. Нейронные сети: полный курс, 2-е изд., испр. / С. Хайкин. - М.: Издательский дом "Вильямс", 2006. - 1104с.*
6. *Шамаев С.И. Методы, процедуры и средства аэрокосмической компьютерной радиотомографии приповерхностных областей Земли*
7. *Тихонов О решении некорректно поставленных задач*
8. *Исиченко И.В., Коновалов А.В., Левченко Е.С., Савин А.С. Обратная задача обтекания особенностей плоским потоком идеальной жидкости со свободной границей // ПМТФ. – 1989. – № 6 – С.86 – 91.*
9. *Manning, C. Introduction to Information Retrieval / C. Manning, P. Raghavan, H. Schutze. - Cambridge University Press, 2008. - 544 p.*
10. *Max Kuhn, Kjell Johnson. Applied Predictive Modeling // Springer, 2013*
11. *Коновалов А.В., Левченко Е.С., Савин А.С. Восстановление плоского течения тяжелой идеальной жидкости по форме ее свободной поверхности // Докл. АН СССР. – 1989.– 305, № 2. – С. 294 – 296.*
12. *Havelock T.H. The forces on a circular submerged in a uniform stream // Proc. Roy. Soc. London 1936. A 157, № 892. С. 526 – 534.*
13. *Nathan M, James W. Big Data: Principles and best practices of scalable realtime data systems: Manning Publications. 2015. 325 c.*
14. *Себастьян Рашка.* Python и машинное обучение. ДМК-Пресс, 2017.

С. 11 –408.

1. *Сретенский Л.Н.* Теория волновых движений жидкости. М.: Наука, 1977. 815с.
2. *Рышард Т.* Элементарное введение в технологию нейронных сетей с примерами программ. М.: Физматлит, 2011. 408 c.
3. *Галушкин А.И.* Нейронные сети. Основы теории. М.: Горячая линия. Телеком, 2010. 450 с.
4. *Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач.– М.: Наука, 1979. – 286 с.*

|  |
| --- |
|  |