Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Кафедра интеллектуальных информационных технологий

Оценка работы \_\_\_\_\_\_\_\_\_\_

Руководитель от УрФУ \_\_\_\_\_\_\_\_\_\_

**РЕШЕНИЕ ЗАДАЧИ СИМВОЛЬНОЙ РЕГРЕССИИ ПОСРЕДСТВОМ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ**

ОТЧЕТ

по научно-исследовательской практике

Руководитель от предприятия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[Основные сведения о компании 3](#_Toc472581843)

[ВВЕДЕНИЕ 4](#_Toc472581844)

[Информация о приложении 5](#_Toc472581845)

[Концепция 5](#_Toc472581846)

[Функционал приложения 6](#_Toc472581847)

[Использованные инструменты разработки 7](#_Toc472581848)

[PyCharm 7](#_Toc472581849)

[Использованные технологии 9](#_Toc472581850)

[Git 9](#_Toc472581851)

[GitHub 10](#_Toc472581852)

[Trello 10](#_Toc472581853)

[lxml 11](#_Toc472581854)

[Логирование информации 12](#_Toc472581855)

[Основные сведения 12](#_Toc472581856)

[Практическое применение 13](#_Toc472581857)

[ЗАКЛЮЧЕНИЕ 14](#_Toc472581858)

[СПИСОК ЛИТЕРАТУРЫ 15](#_Toc472581859)

# Основные сведения о компании

Сфера деятельности компании - "Интернет / Связь / Информационные технологии".

Компания "FT-SOFT, компания по разработке мобильных приложений" размещена в каталоге в разделах: Разработка программного обеспечения для мобильных устройств города Екатеринбурга.

Компания занимается разработкой, тестированием и сопровождением мобильных и web приложений. Компания начала свою деятельность в 2008 году и сейчас продолжает активно развиваться.

Среди разработанных компанией приложений можно выделить приложения для пользователей банков, жителей различных жилых комплексов, приложения для сферы медицинских услуг.

# ВВЕДЕНИЕ

Научно-исследовательская практика магистратов является обязательной составной частью образовательной программы подготовки магистров.

Цель научно-исследовательской практики состоит в формировании у магистрантов навыков и умений, необходимых для организации и проведения научных исследований, связанных с подготовкой и написанием магистерской диссертации

Задание на практику

Реализовать систему ведения логов для генетического программирования.

# Информация о приложении

## Концепция

Символьная регрессия – задача нахождения формулы, которая описывает некую зависимость. Более формально, её можно сформулировать как построение регрессионной модели в виде функциональной зависимости.

Регрессионная модель *f(w, x)* – это параметрическое семейство функций, задающее отображение, представленное формулой (1).

|  |  |
| --- | --- |
|  | (1) |

где - пространство параметров;

- пространство свободных переменных;

*Y* - пространство зависимых переменных.

Функциональная зависимость будет строиться посредством генетического программирования. Данный оптимизационный стохастический алгоритм является модификацией генетического алгоритма. Основное отличие генетического программирования заключается в работе со структурами данных переменного размера. Суперпозиция заданных функций будет представлена в виде дерева, преобразующегося в польскую запись для вычисления математического выражения.

Для решения задачи будет случайным образом создана популяция таких деревьев. Узлы дерева являются математическими функциями, а листы терминальными символами, т.е. константами или независимыми переменными.

Каждое дерево – решение проблемы – будет вычисляться для набора свободных переменных. Если полученный результат достаточно точно совпадет с ожидаемым результатом – зависимой переменной, то необходимая функциональная зависимость считается найденной. В противном случае поиск продолжается дальше.

Следует отметить, что мы заранее не задаем вид и размер необходимой функции. Она меняется случайным образом в ходе выполнения программы путем применения различных генетических операций. Поэтому в результате может получиться достаточно сложная функциональная зависимость, которую впоследствии можно упрощать и улучшать другими методами.

## Функционал приложения

1. Загрузка начальной выборки данных из внешнего источника.
2. Указание минимальной необходимой точности (в целом для всей функции и для каждой точки в отдельности).
3. Определение критерия остановки программы (если не достигли промежуточной точности через какое-либо количество итераций, то прекращаем работу программы).
4. Выбор функции приспособленности.
5. Выбор максимальной глубины дерева, вероятностей репродукции, скрещивания и мутаций.
6. Вывод промежуточных и конечных результатов в виде дерева, польской нотации, графика функции и аналитически.

# Использованные инструменты разработки

## PyCharm

PyCharm – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA. PyCharm работает под операционными системами Windows, Mac OS X и Linux.

Выпускается в двух вариантах – бесплатная версия PyCharm Community Edition и поддерживающая больший набор возможностей PyCharm Professional Edition. PyCharm выполняет инспекцию кода на лету, автодополнение, в том числе основываясь на информации, полученной во время исполнения кода, навигацию по коду, обеспечивает множество рефакторингов.

Ключевые возможности:

1. Мощный и функциональный редактор кода с подсветкой синтаксиса, авто-форматированием и авто-отступами для поддерживаемых языков.
2. Простая и мощная навигация в коде.
3. Помощь при написании кода, включающая в себя автодополнение, авто-импорт, шаблоны кода, проверка на совместимость версии интерпретатора языка, и многое другое.
4. Быстрый просмотр документации для любого элемента прямо в окне редактора, просмотр внешней документации через браузер, поддержка docstring – генерация, подсветка, автодополнение и многое другое.
5. Большое количество инспекций кода.
6. Мощный рефакторинг кода, который предоставляет широкие возможности по выполнению быстрых глобальных изменений в проекте.
7. Полная поддержка свежих версий Django фреймворка.
8. Поддержка Google App Engine.
9. Поддержка IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK и др.
10. Поддержка Flask фреймворка и языков Mako и Jinja2.
11. Редактор Javascript, Coffescript, HTML/CSS, SASS, LESS, HAML.
12. Интеграция с системами контроля версий (VCS).
13. UML диаграммы классов, диаграммы моделей Django и Google App Engine.
14. Интегрированное Unit тестирование.
15. Интерактивные консоли для Python, Django, SSH, отладчика и баз данных.
16. Полнофункциональный графический отладчик (Debugger).
17. Поддержка схем наиболее популярных IDE/редакторов. таких как Netbeans, Eclipse, Emacs, эмуляция VIM редактора.
18. Поддерживаемые языки: Python (2.x, 3.x), Jython, Cython, IronPython, PyPy, Javascript, CoffeScript, HTML/CSS, Django/Jinja2 templates, Gql, LESS/SASS/SCSS/HAML, Mako, Puppet, RegExp, Rest, SQL, XML, YAML.
19. PyCharm имеет несколько цветовых схем, а также настраиваемую подсветку синтаксиса кода.
20. Интеграция с баг/issue-треккерами, такими как JIRA, Youtrack, Lighthouse, Pivotal Tracker, GitHub, Redmine, Trac.
21. Огромная, постоянно пополняемая коллекция плагинов.
22. Кросс-платформенность (Windows, Mac OS X, Linux).

# Использованные технологии

## Git

Система контроля версий (СКВ) – это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

СКВ даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое. Вообще, если, пользуясь СКВ, вы всё испортите или потеряете файлы, всё можно будет легко восстановить. Вдобавок, накладные расходы за всё, что вы получаете, будут очень маленькими.

Главное отличие Git'а от любых других СКВ (например, Subversion) – это то, как Git смотрит на свои данные. В принципе, большинство других систем хранит информацию как список изменений для файлов. Эти системы (CVS, Subversion, Perforce, Bazaar) относятся к хранимым данным как к набору файлов и изменений, сделанных для каждого из этих файлов во времени.

Git не хранит свои данные в таком виде. Вместо этого Git считает хранимые данные набором слепков небольшой файловой системы. Каждый раз, когда вы фиксируете текущую версию проекта, Git, по сути, сохраняет слепок того, как выглядят все файлы проекта на текущий момент. Ради эффективности, если файл не менялся, Git не сохраняет файл снова, а делает ссылку на ранее сохранённый файл.

Это важное отличие Git'а от практически всех других систем контроля версий. Из-за него Git вынужден пересмотреть практически все аспекты контроля версий, которые другие системы переняли от своих предшественниц. Git больше похож на небольшую файловую систему с невероятно мощными инструментами, работающими поверх неё, чем на просто СКВ.

## GitHub

GitHub – веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc.

Создатели сайта называют GitHub «социальной сетью для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории – GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева.

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

## Trello

Trello – это бесплатное веб-приложение для управления проектами небольших групп, разработанное Fog Creek Software.

Trello использует парадигму для управления проектами, известную как канбан, метод, который первоначально был популяризирован Toyota в 1980-х для управления цепочками поставок. Trello использует freemium-бизнес-модель, платные услуги были запущены в 2013 году.

Основным преимуществом Trello можно назвать возможность видеть несколько одновременно запущенных проектов и их состояние в текущий момент времени. Если вы руководите группой разработчиков или иных исполнителей, работающих над проектами с конечной датой выполнения или фиксированной целью, то данная система может дать вам представление о ходе проектов в любой момент времени.

Важно заметить, что все члены рабочей группы видят в реальном времени изменения, вносимые в проект, и могут наблюдать состояния друг друга так же в реальном времени – онлайн или офлайн, смотрит на проект или нет в настоящий момент.

## lxml

lxml это быстрая и гибкая библиотека для обработки разметки XML и HTML на Python. Кроме того, в ней присутствует возможность разложения элементов документа в дерево.

В то время как для простого DOM- или SAX-анализа небольших файлов достаточно любой XML-библиотеки, разработчики все чаще сталкиваются с крупными наборами данных и необходимостью разбирать XML в реальном времени. В то же время опытные XML-разработчики могут предпочесть более XML-ориентированные языки, такие как XPath или XSLT, за их лаконичность и выразительность. Идеальным выходом из этой ситуации было бы иметь доступ к декларативному синтаксису XPath, сохраняя при этом общую функциональность, доступную в Python.

lxml – первая XML-библиотека для Python, которая имеет высокую производительность и содержит встроенную поддержку XPath 1.0, XSLT 1.0, пользовательские элементарные классы и даже типичный для Python интерфейс привязки данных. Библиотека lxml построена на базе двух библиотек языка С: libxml2 и libxslt. Они обеспечивают большую часть возможностей для основных задач анализа, сериализации и преобразования.

Для обработки XML и HTML можно использовать и BeautifulSoup, но эта библиотека использует слишком много памяти и не удовлетворяют требованиям по скорости для больших файлов. Библиотека lxml очень быстро открывает и обрабатывает большие XML или HTML файлы, поэтому была выбрана для применения в данной программе.

# Логирование информации

## Основные сведения

Довольно часто возникает необходимость в анализе текущего состояния системы. Для этого может использоваться журнал событий, ведение которого носит название – логирование, а события называют логами.

Логирование может осуществляться разными путями:

1. Самый простой и потому самый распространенный способ – это логирование в текстовый файл. Способ, при котором отдельное событие представляет собой отдельную строку. Способ хорош как с точки зрения реализации – довольно легко наладить такое логирование в коде большинства языков программирования, – так и со стороны использования – читать такой лог можно любым текстовым редактором.
2. Чуть более сложный случай – логи, в которых отдельное событие представляет собой не одну строку, а несколько. С некоторым допущением к этому же типу относятся логи, которые пишутся в формате XML (либо сходных с ним форматах данных). Такой лог гораздо более сложен для анализа, потому что каждое событие может представлять собой набор более мелких записей. Для чтения таких логов чаще всего используется какое-то специально ПО, так как лог, в котором каждое событие растянуто на несколько строк, а еще и сами события зависят друг от друга, довольно тяжело интерпретировать.
3. Бинарный лог представляет собой самый нечитаемый тип логов. Для того чтобы с ними работать, нужна специальная программа (обычно от того же производителя, что и приложение, которое пишет такой лог), с помощью которой бинарный лог и анализируется. Обычно бинарный лог — это последовательно сбрасываемые в файл структуры, которые разделяются символом-разделителем. Обрабатывать такой лог очень тяжело, впрочем, довольно часто в технической информации, которую предоставляет производитель, есть описание структуры такого лога.
4. Приложения, которые используют базы данных либо сами являются СУБД, довольно часто используют базу данных в качестве хранилища логов. В большинстве своем это отдельная таблица базы данных, каждая строка которой является отдельным событием. Такое логирование часто может отрицательно сказаться на общей производительности базы данных, так как логирование в базу данных может быть довольно интенсивным.

## Практическое применение

Генетическое программирование предполагает работу со сложными структурами данных – деревьями. В связи с этим операции с подобными элементами занимают большое количество времени. Также для успешного завершения работы программы – нахождения оптимального решения – необходимо работать с большой популяцией таких особей – деревьев, что в свою очередь оказывает влияние на время работы программы.

По вышеперечисленным причинам работа программного комплекса длится большой промежуток времени. Есть необходимость анализировать работу программы, наблюдать за промежуточными вычислениями. Для этой цели было решено вести журнал событий программы с записью в отдельный файл формата XML.

Для полноценного анализа вычислений программы необходимо знать начальное состояние системы, которое определяется следующими параметрами: функциональное и терминальное множества, размер начальной и рабочей популяций, вероятности репродукции, кроссинговера и мутации, максимальную допустимую погрешность и глубину дерева.

Кроме входных данных нужно записывать промежуточные состояния системы, такие как сведения о текущей популяции, наиболее приспособленный индивид и его погрешность, информация о родителях полученных особей и т.д.

# ЗАКЛЮЧЕНИЕ

Достоинством использования метода генетического программирования для решения задачи установления функциональная зависимости данных является точность совпадения полученной функции в точках выборки. Если у нас есть дополнительная информация о функции кроме ее точек, мы легко можем использовать ее при применении операции репродукции. Концептуальная простота генетического программирования также является важным преимуществом его практического применения.

В ходе прохождения практики была разработана система ведения журнала событий программного комплекса. Полученные данные позволили провести анализ состояний системы, привели к улучшению работы программы. В связи с этим можно считать разработанную систему успешно внедренной в программный комплекс.

# СПИСОК ЛИТЕРАТУРЫ

1. Koza, John R. Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA, USA, 1992.
2. Langdon, William B. Genetic programming and data structures. Department of Computer Science, University College London, 1996.
3. Mitchell Melanie. An introduction to Genetic Algorithms. MIT Press, Cambridge, London, 1998.
4. Xinjie Yu, Mitsuo Gen. Introduction to Evolutionary Algorithms. Springer, London Limited 2010.
5. Библиотека lxml [Электронный ресурс]. URL: http://lxml.de/ (дата обращения – 01.09.2016)