**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Прикладная математика и информатика»

**Отчет о программном проекте**

на тему "Обучающие материалы отечественной нейросетевой платформы PuzzleLib"

(итоговый, этап 3)

**Выполнил**:

студент группы БПМИ 198 В. В. Вьялков



Подпись И.О. Фамилия

08.02.2021

Дата

**Принял**:

руководитель проекта Дмитрий Валерьвевич Пантюхин \_

Имя, Отчество, Фамилия

Старший преподаватель \_

Должность, ученое звание

Департамент программной инженерии \_

Место работы (Компания или подразделение НИУ ВШЭ)

|  |  |  |
| --- | --- | --- |
| Дата проверки 2021 |  |  |
|  | Оценка | Подпись |
|  | (по 10-тибалльной шкале) |  |

**Москва 2021**

# Реферат

Проект направлен на то, чтобы разработать понятную и доступную новчикам в машинном обучении документацию для библиотеки PuzzleLib и пополнить функционал новыми модулями.

Для успешной реализации проекта были изучены методы машинного обучения, проанализированы аналоги, выделены слабые стороны библиотеки PuzzleLib и сильные стороны конкурентов. Изучены материалы по оформлению документаций.

Цель и задачи проекта были выполнены: библиотека пополнилась новыми архитектурами, а документация была переработана

# Содержание

[Реферат 2](#_Toc73210821)

[Содержание 3](#_Toc73210822)

[Основные термины и определения 4](#_Toc73210823)

[Введение 5](#_Toc73210824)

[Сравнительные анализ источников и аналогов 7](#_Toc73210825)

[Описание требований к программному проекту 9](#_Toc73210826)

[Реализация проекта 10](#_Toc73210827)

[Описание подходов к реализации 10](#_Toc73210828)

[Обоснованный выбор технологий и средств разработки 11](#_Toc73210829)

[Список источников 12](#_Toc73210830)

# Основные термины и определения

**Программирование** — раздел информатики, область знаний об алгоритмах и программах и их свойствах, а такжеисполнителях алгоритмов и программ

**Машинное обучение (Machine Learning)** - класс методов, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений большого количетва похожих задач.

**Нейронная сеть (Neural Network)** — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

**Библиотека** – набор функций, позволяющих собрать и обучить нейронную сеть без потребности в понимании математической реализации модулей

**Архитектура сети** - совокупность значимых параметров сети, определяющих её назначение и возможности, среди которых: структура и конфигурация слоев нейронной сети а также характер взаимодействия между слоями.

**Google Colaboratory** - облачная платформа с предустановленным программным обеспечением и библиотеками для проведения вычислений, которую удобно использовать в том числе для обучения глубоких нейронных сетей.

**Jupyter Notebook** — это командная оболочка для интерактивных вычислений, позволяющая создавать файлы с разметкой и кодом. Аналог Google Colaboratory

**Пайплайн обучения –** последовательные стадии подготовки и преобразования данных перед загрузкой в модель

# Введение

С появлением программирования человечество научилось решать множество ранее нерешаемых задач. Знания об алгоритмах, языки программирования и сами комьютеры (с постоянным увеличением вычислительных мощностей), позволили оптимизировать множество процессов.

Прорывным направлением программирования в наши дни является использование машинного обучения для решения прикладных задач. Первые нейронные сети, появились еще в XX веке, а в XXI темпы развития этой отрасли науки значительно ускорились. Сейчас нейронные сети способны решать задачи прогнозирования, классификации, оптимизации, генерации и многие другие.

Программирование нейронных сетей зачастую предполагает использование готовых библиотек с заранее определенными модулями. В наше время существует большое количество инструментов для машинного обучения (Tensor Flow, Keras, Torch и др). Различия между ними прежде всего определяется количеством доступных модулей, быстродействием полученных программ и удобством использования библиотеки программистами.

Библиотека PuzzleLib – набор программных модулей для использования в машинном обучении, полностью разработанный в России. В некоторых задачах этот инструмент превосходит по быстродйествию зарубежные аналоги, но проигрывает в количестве функционала и понятности документации. Ключевыми задачами компании, отвечающей за разработку и поддержку этой библиотеки, являются дополнение библиотеки новыми модулями и написание понятной документации. Мотивация данного проекта происходит из задач компании.

**Цель проекта:** разработать понятную документацую для библиотеки Puzzlelib и расширить функционал новыми модулями

**Задачи проекта:**

* Изучить методы машинного обучения и их применение для решения прикладных задач
* Проанализировать документацию PuzzleLib, выявить слабые стороны, разработать методы для повешния удобства использования библиотеки через переработку и дополнение документации
* Проанализировать фукнционал библиотеки и предложить модули для его расширения
* Разработать новые модули для библиотеки PuzzleLib
* Провести переработку библиотеки, повысить читаемость, исравить ошибки, сделать ее более понятной

# Сравнительные анализ источников и аналогов

В качестве источников для реализации задач проекта были выбраны платформа Coursera, несколько специализированных сайтов по машинному обучению и сайты с документацей зарубежных аналогов. (см. Список источников).

Стоит отметить, что на платформе Coursera, курсы «Введение в машинное обучение» и «Машинное обучение и анализ данных» понятным языком объясняют устройство нейронных сетей. Этот ресурс поможет в освоении машинного обучения прежде всего тем, кто до этого не сталкивался с этой отраслью.

Из аналогов библиотеки PuzzleLib стоит выделить только зарубежные, разработок подобной сложности в России найдено не было. Рассмотрим ключевые аналоги: TensorFlow, Keras, Pytorch.

Для оцени библиотек были выработаны следующие критерии сравнения:

* Наличие и качество документации
* Быстродействие
* Качество поддержки

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий сравнения** | **TensorFlow** | **Keras** | **Pytorch** |
| Наличие и качество документации | Компнаия Google снабдила библиотеку понятной документацией, но на уровне тех, кто уже хорошо понимает в разработке | Средней понятности документация, похоже на TensorFlow, есть гайды с разбором конкретных задач | Самая понятная и подробная документация из представленных |
| Быстродействие (По приоритету) | 2 | 3 | 1 |
| Качество подержки | Хорошее, большое количество материвалов информации на форумах | Хорошее, большое количество материвалов информации на форумах, большое комьюнити | Большое комьюнити, на StackOverflow разрабрано много проблем |

# Описание требований к программному проекту

На основе анализа источников и аналогов были выработаны функциональные и нефункциональные требования для реализации проекта.

**Функциональные требования**

* Взаимодействие пользователя с программным кодом (Интерактивное)
* Реализация архитектур и пайплайна обучения
* Реализация доступных и актуальных для прикладных задач примеров для библиотеки
* Программный код должен быть снабжен подробными комментариями на русском языке
* Программный код должен быть совместим с библиотекой PuzzleLib

**Нефункциональные требования**

* Взаимодействие пользователя с программным кодом реализовано через Google Colaboratory или Jupyter Notebook
* Документация должна быть написана без ошибок, понятным и доступным для неподготовленного пользователя языком
* Во время исполнения кода из примеров не должно возникать багов

# Основные методы и алгоритмы

## Densenet

Особенности архитектуры DensеNet в том, что вместо того, чтобы делать сеть шире и глубже, автор предлагает переиспользование feature maps со всех слоев.

DenseNet вместо композитных операций суммирования на выходных feature map’ах использует их конкатенацию, что позволяет добиваться высокой точности при меньшем кол-все весов.

Максимальный информационный и градиентный flow обеспечивается за счёт того, что каждый слой соединятся напрямую с каждым

## GAN

Суть работы GAN заключается в использовании двух нейронных сетей – генератора, непосредственного создающего изображения и дескриминатора, оценивающего их подленность.

Задача генератора – сгенерировать изображение, которые дескриминатор распознает как принадлежащее датасету, а дескриминатора – понять, дают ему картинку из тренеровочной выборки или нет.

* Генератор получает на входе случайное число и отдаёт изображение.
* Это изображение подается в дискриминатор вместе с потоком изображений, взятых из фактического набора данных.
* Дискриминатор принимает все изображения и определяет вероятности: числа от 0 до 1. 1 представляет собой подлинное изображение, а 0 - фальшивое.

## Документация

Методы разработки документации были выработаны после анализа конкурентов. Компания, по запросу которой реализуется текущий проект, ставила акцент на то, что обучающие материалы должны быть понятны школьнику.

Было принято решение протестировать полученную с учетом практик зарубежных аналогов документацию на выборке из школьников учебных заведений с углубленным изучением информатики, собрать обратную связь и внести правки в документацию.

# Реализация проекта

## Описание подходов к реализации

В рамках группового проекта задачи были поделены на несколько частей. Перечень задач, глобально рассмотренных в требованиях к программному проекту, непосредственно закрепленных за мной, ограничивается следующими:

* Реализация архитектуры и пайплайна обучения сети DenseNet
* Реализация архитектуры и пайплайна обучения одного из типов сетей GAN
* Написание документации для некоторых частей раздела Modules и Optimizers

Документация и реализация представляют собой файлы с кодом и пояснениями к нему, с возможностью исполнения кода и экспериментов с ним. Основная задача: предоставить пользователю возможность взаимодействовать с библиотекой на этапе изучения и сделать код понятным для новичков.

Чтобы повысить понятность программ для тех, кто только начинает изучать нейронные сети, код будет содержать комментарии на русском языке в тех местах, где это необходимо для лучшего понимания.

Для проверки качества документации планируется собрать выборку из школьников учебных заведений с углубленным изучением информатики и протестировать полученные материалы на них. Далее будут выделены наименее понятные моменты, на основе полученных данных реализацию архитектур и документацию планируется доработать.

## Обоснованный выбор технологий и средств разработки

Для реализации архитектур нейронных сетей с пайплайном обучения и дополнения документации библиотеки PuzzleLib был выбрал инструмент Jupyter Notebook. Это среда разработки, позволяющая исполнять блоки кода и вставлять в разметку страницы блоки с текстом. Данный функционал позволяет оформить учебные материалы в виде интерактивной статьи, что является очень удобным форматом в контексте задач проекта.

Инструмент Jupyter Notebook был выбран из удобства использования (ранее использовался мной для написания кода). Рассматривался вариант использовать аналог Google Colaborate – аналог выбранного инструмента, но реализованный в веб-интерфейсе (нет необходимости устанавливать программу на компьютер). В силу того, что формат файлов из Jupyter Notebook полностью совместим с Google Colaborate, каждый пользователь может открыть учебные материалы в удобном ему редакторе.

Библиотека PuzzleLib написана на языке Python, поэтому дополнительные модули и код в документации должен быть написан на этом языке из соображений совместимости.

Для успешной реализации поставленных задач необходимо было освоить методы машинного обучения и проанализировать документацию аналогов библиотеки PuzzleLib (Для анализа были выбраны PyTorch, TensorFlow, Keras)

## Ссылка на репозиторий и ссылка для запуска

<https://github.com/BlackDesk/HSE_PuzzleLib>

## Особенности запуска

Чтобы запустить программу, необходимо скачать файлы с репозитория на GitHub (<https://github.com/BlackDesk/HSE_PuzzleLib>), и открыть их в любой из программ Google Colaborage или Jupyter Notebook. Открытые файлы представляют собой интерактивные статьи, исполнять код можно прямо внутри файла через интерпретатор среды разработки.

# Основные результаты и перспективы дальнейшей работы

Резльтаты работы в контектсе задач, закрепленных за мной, можно разделить на две части:

* Были переработаны некоторые блоки документации PuzzleLib, исправлены недочеты, разработаны новые примеры и понятные пояснения к ним
* Реализованы архитектуры Densenet и GAN с пайплайном обучения и поясняющими материалами.

Задачи проекта были выполнены, а цель достигнута.

Перспективы дальнейшей работы:

* Расширения функционала библиотеки новыми архитектурами нейронных сетей
* Продолжение переработки документации и создание туториалов

Библиотека PuzzleLib обладает широким функционалом, требующим объемной документации и большого количества обучающих материалов. В связи с этим в перспективе возможна дальнейшая работа над улучшением библиотеки.

# Список источников

1. Pytorh Documentation [Электронный ресурс]. <https://pytorch.org/docs/stable/index.html> Режим доступа: Свободный (дата обращения: 22.12.2020)
2. TensorFlow Documentation [Электронный ресурс]. <https://www.tensorflow.org/api_docs> Режим доступа: Свободный (дата обращения: 22.12.2020)
3. PuzzleLib Documentation [Электронный ресурс].   
   <https://puzzlelib.org/ru/> Режим доступа: Свободный (дата обращения: 22.12.2020)
4. Keras Documentation [Электронный ресурс].   
   <https://keras.io/> Режим доступа: Свободный (дата обращения: 22.12.2020)
5. Coursera, Введение в машинное обучение [Электронный ресурс].   
   <https://ru.coursera.org/learn/vvedenie-mashinnoe-obuchenie> Режим доступа: Ограниченный (дата обращения: 22.12.2020)
6. Coursera, Машинное обучение и анализ данных [Электронный ресурс].   
   <https://ru.coursera.org/specializations/machine-learning-data-analysis> Режим доступа: Ограниченный (дата обращения: 22.12.2020)
7. Habr, Сравнение фреймворков для машинного обучения [Электронный ресурс]  
   <https://habr.com/ru/company/otus/blog/443874/> Режим доступа: Свободный (дата обращения: 22.12.2020)
8. Термины и опредления в машинном обучении [Электронный ресурс]  
   <https://neural-university.ru/neural-networks-basics#rec116464826> Режим доступа: Cвободный (дата обращения: 22.12.2020)