

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра «Компьютерные системы и программные технологии»

КУРСОВОЙ ПРОЕКТ

Название темы

по дисциплине «Технологии программирования (Java)»

Выполнил
студент гр.13531/2

С.Н. Полухин

Руководитель

А.О. Алексюк

«___» _____ 2018 г.

Санкт-Петербург

2018

Исходные данные к работе

Back-end (Server and Education):

IDE: IntelliJ IDEA 2018.1.2 (Ultimate Edition)

JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

macOS 10.13.4

Version Java: 10

Библиотеки, не входящие в стандартный набор:

Thumbnails (<https://github.com/coobird/thumbnailator>)

JSON Simple (<https://code.google.com/archive/p/json-simple/>)

Front-end (Client)

IDE: Android Studio 3.1.2

JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Mac OS X 10.13.4

Version Java: 8

Библиотеки, не входящие в стандартный набор:

JCraft (<http://www.jcraft.com/jsch/>)

Текст программы

Server — <https://github.com/poluh/RecipeServer>

Client — <https://github.com/poluh/RecipeClient>

Описание программы

Server:

Main.java

Принимает в виде аргумента путь до картинки, которую необходимо обработать. Выводит на консоль абстрактную информацию о своей работе. Формирует JSON ответ.

logic package:

image:

geometry:

Point.java

Класс точки. Что с него взять. Считает дистанцию от точки до точки.

ImagePreprocessor.java

Преобразовывает изображение в массив нулей и единиц, каждый обозначающий белый/черный цвета.

Для этого преобразовывает изображения в оттенки серого, бинаризирует оное, выбирая порог по гистограмме и вычисляя яркости пикселей, обрезает (в одно или несколько изображений), а затем делает resize до квадрата 50x50 пикселей, преобразовывая в последствии в массив нулей и единиц.

json:

JSON.java

Упрощает работу с API сторонней библиотеки.

network:

Layer.java

Хранит список нейронов конкретного слоя.

Neuron.java

Нейрон. Имеет вес и сигмоидальную функцию активации

Network.java

Хранит список всех слоев, на основе которых происходит вычисление результата. Получает на вход изображение, которое обрабатывает в несколько потоков в нескольких слоях одновременно, создавая список из наиболее точного и вторичных результатов.

Education:

Education.java:

Принимает на вход путь до обучающих изображений, на основе которых в несколько потоков обучает сеть для каждой цифры.

Client:

com.client:

image:

Point.java

Точка.

BitmapPreprocessor.java

Сжимает изображение для более быстрой передачи на сервер.

server.connetion:

ServerForwarder.java:

Принимает на вход файл-изображение, отправляет оное на сервер. Так же получает ответ от сервера.

UserInfo.java

Хранит информацию для подключение к серверу.

UI:

Drawing.java:

Создает контейнер-canvas, на котором можно рисовать, обрабатывая полученное изображение через **BitmapPreprocessor** и локально сохраняя. Обрабатывает ответ сервера и выводит на экран через **Shower**

MainActivity.java:

Создает главное окно приложения.

Shower.java:

Выводит на экран ответ от сервера в человеческом виде, а так же показывает, как нейросеть «видит» изображение.

Испытание программы

В составе Server имеются unit-тесты, которые проверяют сеть на выборке из пятисот изображений. По результатам, выводится процент ошибки. (На текущий момент — 5%). Client тестировался в ручном режиме, путем бесконечного рисования в приложении.

Заключение

В заключении хочется подвести итог: данная работа заняла бы у меня гораздо меньше времени, не будь я столь ленив. Но самое важное, пожалуй, что я получил — опыт. Бесценный опыт работы с новыми, неизвестными мне технологиями. Я практически впервые познакомился с работой над приложением под Android. А так же с нуля собрал собственную нейросеть и сделал возможность работы через сервер.

UI приложений в течении первого курса я написал достаточно, но с подобными сложностями встреч еще провести не удавалось.

Изначально работа велась в куда меньших масштабах. Но сейчас, глядя на уже созданное, я понимаю, насколько много мне предстоит еще сделать. В моих мечтах — анализатор рукописного текста. Для начала, хотя бы английского. Надеюсь, что смогу сделать это в течении года, а пока представляю вам это — анализатор рукописных чисел.

Предстоит еще много исследований интернета и чтения статей, но данное приложение — небольшой шаг к этому.

Хочу отдельно выразить благодарности Артему Олеговичу, за помощь в разработке и работе с сервером, а так же Инге Егоровой и Ахину Марату, за помощь на старте на первом семестре. А так же своей жене, за то, что вовремя меня бесила и успокаивала.

Список используемых источников

Documentation of Java (<https://docs.oracle.com/>)

StackOverFlow (<https://stackoverflow.com>)

А так же статьи на хабре. В частности:

Введение в нейросети (<https://habr.com/post/342334/>)

Введение в архитектуры нейронных сетей (<https://habr.com/company/oleg-bunin/blog/340184/>)

Нейронные сети: практическое применение (<https://habr.com/post/322392/>)

Алгоритм машинного обучения Flappy (<https://habr.com/post/336612/>)

