

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОЗДАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ
ПОРОД СОБАК НА ОСНОВЕ НЕЙРОННОЙ СЕТИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Шелухина Андрея Павловича

Научный руководитель

Ст. преподаватель

М. И. Сафрончик

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2022

ВВЕДЕНИЕ

Информационные технологии развиваются огромными темпами и проникают во все большие сферы деятельности, охватывая различные аспекты жизни. Сейчас сложно представить человека, который не пользуется мобильными приложениями, программами для обработки фотографий или же мессенджерами. Среди крупных IT-компаний происходит борьба за инновационность и упрощение взаимодействия пользователя с их продуктами, в этом им помогают, в том числе, нейронные сети. В научных отраслях информационные технологии необходимы как никогда раньше, ведь ни одно исследование сейчас не сможет обойтись без нейронных сетей, выполняющих сложные вычисления и моделирование. В индустрии развлечений, с помощью нейронных сетей, создается компьютерная графика, анимация и даже музыка. Они повсеместно используются для поиска объекта по картинке, сортировки данных, компьютерного зрения в электромобилях.

Целью дипломной работы является — разработать мобильное приложение, позволяющее распознать породу собаки по фото на основе нейронной сети, решающей задачу классификации. Для достижения цели необходимо выполнить следующие задачи:

1. исследовать информацию об искусственном интеллекте и нейронных сетях;
2. разработать нейронную сеть для распознавания пород собак;
3. обучить данную нейронную сеть на выбранном датасете;
4. разработать главное меню приложения;
5. создать отдельную страницу со списком из пород собак;
6. создать страницы для каждой породы собак и связать их между собой;
7. разработать кнопки вызова камеры и галереи смартфона в главном меню приложения.

Структура и объем работы. Для решения поставленных задач выполнена выпускная квалификационная работа, которая включает в себя введение, 3 основные главы, заключение, список использованных источников из 20 наименований и 5 приложений. Работа изложена на 88 страницах, содержит 18 рисунков.

Первая глава имеет название «Структура и принцип работы нейронных сетей» и содержит информацию об основных понятиях в теме нейронных се-

тей, описание их внутреннего устройства. Также в главе были описаны внутренние компоненты нейронной сети, которые были использованы в разработке. В конце главы была описана сама модель нейронной сети, по которой в дальнейшем была создана нейронная сеть.

Вторая глава имеет название «Создание нейронной сети для распознавания породы собаки» и содержит информацию об алгоритме создания и обучения нейронной сети.

Третья глава имеет название «Создание мобильного приложения для распознавания породы собак» и содержит подробное описание, как создавалось мобильное приложение на базе нейронной сети.

Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложениями с кодом А-Д.

Основное содержание работы

Объект исследования. Была поставлена задача разработать мобильное приложение на базе нейронной сети по распознаванию пород собак.

Структура и принцип работы нейронных сетей. Компьютерная нейронная сеть является моделью нейронных сетей мозга живого организма. В компьютерных нейронных сетях работают тех же принципы организации и функционирования.

Компьютерная нейронная сеть состоит из трех слоев:

- Входной слой.
- Скрытый(Вычислительный) слой.
- Выходной слой.

Именно в скрытом слое находятся нейроны, на которых построен весь принцип работы нейронных сетей. С математической точки зрения искусственный нейрон — это сумматор всех входящих сигналов, применяющий к полученной взвешенной сумме некоторую простую, в общем случае, нелинейную функцию, непрерывную на всей области определения. Обычно, данная функция монотонно возрастает. Полученный результат посылается на единственный выход. Нейроны объединяются между собой определенным образом, образуя искусственную нейронную сеть. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал поступает на синапсы следующих нейронов. Каждый синапс характеризуется величиной синаптической связи, в искусственных нейронных сетях такие связи называют весами. Веса являются аналогом биологических связей между нейронами.

Функция активации и ее виды. Функция активации — это функция, которая добавляется в искусственную нейронную сеть, чтобы помочь сети изучать сложные закономерности в данных. При сравнении с моделью, основанной на нейронах, которая находится в нашем мозге, функция активации в конце решает, что должно быть передано следующему нейрону. Он принимает выходной сигнал из предыдущей ячейки и преобразует его в некоторую форму, которая может быть принята в качестве входного сигнала для следующей

ячейки. На рисунке ниже можно увидеть сравнение биологического нейрона с его математической моделью.

Список самых известных, по тем или иным причинам, функций активации:

- Сигмоида.
- Softmax.
- ReLU.

Функция активации ReLU6. Функция ReLU6 является функцией активации для сверточных нейронных сетей и имеет формулу(1):

$$F(x) = \min(\max(0, x), 6). \quad (1)$$

Основное различие между ReLU и ReLU6 заключается в том, что ReLU допускает очень высокие значения на положительной стороне, в то время как ReLU6 ограничивает значение 6 на положительной стороне. Любое входное значение, которое равно 6 или больше 6, будет ограничено значением 6. Также ReLU6 является нелинейной функцией активацией.

Помимо различий, данная функция имеет очень много общего с функцией ReLU:

- Является отличной альтернативой функциям активации Сигмоида и \tanh .
- У этой функции нет проблемы с исчезающим градиентом.
- Эта функция является вычислительно недорогой. Считается, что сходимость результатов в 6 раз быстрее, чем у сигмовидной и \tanh -функций.
- Если входное значение равно 0 или больше 0, функция ReLU6 выводит входные данные как есть. Если входное значение меньше 0, функция ReLU выводит значение 0.
- Функция ReLU6 состоит из трех линейных компонентов, а ReLU из двух. Однако обе функции являются кусочно-линейными. Функция ReLU6 является нелинейной функцией.
- Сходимость происходит быстрее, чем сигмовидные и \tanh -функции. Это связано с тем, что функция ReLU6 имеет фиксированную производную для одной линейной составляющей и нулевую производную для других 8 линейных составляющих. Таким образом, процесс обучения происходит намного быстрее с помощью функции ReLU6.
- Вычисления выполняются намного быстрее, поскольку в функцию не

включаются экспоненциальные члены.

Функции потерь(Loss function). Функция потерь находится в центре нейронной сети. Она используется для расчета ошибки между реальными и полученными ответами. Наша глобальная цель — минимизировать эту ошибку. Таким образом, функция потерь эффективно приближает обучение нейронной сети к этой цели. Обучение нейронной сети заключается в поиске глобального минимума функции потерь.

Функция потерь измеряет «насколько хороша» нейронная сеть в отношении данной обучающей выборки и ожидаемых ответов. Она также может зависеть от таких переменных, как веса и смещения.

Функция потерь одномерна и не является вектором, поскольку она оценивает, насколько хорошо нейронная сеть работает в целом.

Некоторые известные функции потерь:

- Квадратичная (среднеквадратичное отклонение).
- Кросс-энтропия.
- AdaBoost функция.

Оптимизация нейронных сетей, функции-оптимизаторы. Оптимизатор — это метод достижения лучших результатов, помощь в ускорении обучения. Другими словами, это алгоритм, используемый для незначительного изменения параметров, таких как веса и скорость обучения, чтобы модель работала правильно и быстро. Самые распространенные оптимизаторы нейронных сетей:

- Стохастический градиентный спуск(SGD).
- Среднеквадратичное распространение.
- Оптимизатор Adam.

Модель нейронной сети MobileNetV2. MobileNetV2 — это сверточная нейронная сеть, глубиной в 53 слоя. Можно загрузить предварительно обученную версию сети, обученной больше чем на миллионе изображений от базы данных ImageNet. Данная нейронная сеть может провести классификацию до 1000 классов объектов, таких как клавиатура, мышь, карандаш и многие животные. В результате сеть предобучена и имеет богатые представления функции для широкого спектра изображений. Входной размер изображений для данной модели 224 на 224 пикселя. Сверточная часть сети состоит из одного слоя с 3x3 сверткой в начале и тринадцати блоков с постепенно увеличивающимся

числом фильтров и понижающейся пространственной размерностью тензора. Одной из особенностей данной архитектуры является отсутствие max-pooling слоев. На замену им поставлена свертка, имеющая параметр stride со значением 2. Данная замена снижает пространственную размерность сети. Также в структуре существует два гиперпараметра:

- Параметр α является множителем ширины. От множителя ширины зависит количество каналов в слоях сети. Например $\alpha = 0.25$ определяет архитектуру с уменьшенным в четыре раза числом каналов на выходе каждого блока.
- Параметр ρ является множителем глубины. Множитель глубины отвечает за пространственные размеры входных тензоров. Например, $\rho = 0.5$ будет означать уменьшение вдвое высоты и ширины для feature map, подаваемой на вход каждому слою.

С помощью этих параметров можно контролировать размеры сети: уменьшая α и ρ , можно снизить точность предсказания, но при этом увеличить скорость работы и уменьшить требуемую память.

Как и в MobileNet, в MobileNetV2 существуют сверточные блоки с шагом 1 и с шагом 2. Блоки с шагом 2 предназначены для снижения пространственной размерности тензора и, в отличие от блока с шагом 1, не имеют residual connections.

Создание нейронной сети. Для создания любой нейронной сети необходим подходящий датасет. Был взят датасет «70 Dog Breeds-Image Data Set». Он подходит для проекта по нескольким причинам:

- Большое количество пород собак.
- Достаточное количество фотографий для обучения нейронной сети.
- Фотографии собак с разных ракурсов, что сделает нейронную более устойчивой к ошибкам человека при создании фотографии.
- Цветные фото.
- Фото более хорошего качества в сравнении с другими датасетами данной направленности.

Данная нейронная сеть решает задачу классификации, а так же она должна быть адаптируемой под мобильное приложение. Исходя из этого была выбрана модель MobileNetV2. Модель имеет 4 миллиона параметров, при этом ее веса занимают 13Мб памяти. Для начала нужно убедиться, что датасет корректно

устроен и поэтому осуществляется специальная выгрузка случайных фото с их классами. Используется предобученная модель на датасете ImageNet, что позволяет сократить время обучения. Алгоритм разработки нейронной сети был следующим:

- Используется предобученная модель на датасете ImageNet, что позволяет сократить время обучения.
- Заморозили веса модели, чтобы обучить только последний слой классификации.
- Добавили слой классификации вместо последнего слоя модели. Функция активации - ReLU, так как она хорошо справляется с задачами классификации и отлично подходит для модели нейронной сети с дальнейшей интеграцией в мобильное приложение.
- Определили функцию потерь, функцию оптимизации и скорость обучения. Функция потерь — перекрестная энтропия, которая используется в задачах мультиклассовой классификации. Оптимизатор — Adam часто обеспечивает сходимость быстрее, чем классический градиентный спуск (SGD), что необходимо нам в данном проекте. Скорость обучения: классическая и рекомендованная для Adam — 0.001.
- Описываем функцию обучения модели.
- Выводим результат работы нейронной сети.

Дизайн мобильного приложения.

В основе интерфейса и дизайна стоит концепция минимализма. Главный экран является многофункциональным меню, открывает пользователю список всех действий. Раздел с информацией о породах находится на отдельной странице в виде списка пород. При нажатии на породу из списка пользователя переносит на новую страницу с информацией о породе и о похожих породах. При разработке дизайна были учтены основные правила современного дизайна приложений:

- Один цвет является доминирующим, остальные - нейтральные и неяркие.
- Упор на минимализм, отсутствие теней, объемных объектов.
- Отсутствие слишком ярких цветов для сохранения длительной концентрации.

Разработка навигации в приложении. Для навигации была импортирована специальная вспомогательная библиотека, с помощью которой была

построена навигация в приложении. Все страницы были связаны между собой и реализована простая возможность перехода на другие страницы по созданным кнопкам.

Разработка главного экрана. Главный экран мобильного приложения по своей сути является меню, в котором пользователь может выбрать необходимое ему действие. Всего есть 3 кнопки:

- Загрузить из галереи.
- Сделать фото.
- Список пород.

При нажатии на кнопку «Загрузить из галереи» перед пользователем открывается его галерея фотографий, где он может выбрать необходимую ему фотографию собаки для проверки породы.

При нажатии на кнопку «Сделать фото» перед пользователем открывается камера, с помощью которой пользователь может сделать фотографию собаки для проверки ее породы.

При нажатии на кнопку «Список пород» пользователь попадает на страницу «Породы», которая хранит список всех пород в виде справочника.

Для кнопок «Загрузить из галереи» и «Сделать фото» были реализованы методы по вызову камеры и галереи. Для кнопки «Список пород» была реализована навигация на соответствующую страницу.

Также, в качестве декора, на главный экран было помещено название проекта мобильного приложения «DoggiDoggi».

Разработка экрана со списком пород. Экран «Породы» представляет собой список всех имеющихся пород. Каждый элемент был реализован в виде кнопки, нажав на которую, пользователя направляло на страницу с породой собаки, на которую он нажал во вкладке «Породы». Всего было реализовано 70 интерактивных элементов списка. Помимо перехода на страницу породы собаки была реализована кнопка «Назад», которая возвращала пользователя на главный экран приложения. Также, в качестве декора, на экран «Породы» было помещено название проекта мобильного приложения «DoggiDoggi».

Разработка страниц с породами собак. Для каждой породы собаки была создана страница с информацией о данной породе. Страница состоит из 5-ти главных частей:

- Навигационная часть, позволяющая пользователю с помощью кнопки

«Назад», вернуться на экран «Породы».

- Название породы собаки.
- Фотография с представителем данной породы.
- Список из нескольких важных особенностей породы.
- Интерактивный список похожих пород.

Все части, за исключением навигационной и интерактивного списка с похожими породами собак являются статичными. Навигационная часть сделана по аналогии с предыдущими страницами приложения. Интерактивный список похожих пород работает аналогично списку на экране «Породы». Когда пользователь нажимает на ту или иную похожую породу, его перенаправляет на страницу породы, соответствующей нажатой кнопке.

Внедрение нейронной сети в мобильное приложение. Заключительным этапом проекта является внедрение нейронной сети путем добавления ее как компоненту, по аналогии с предыдущими шагами. Обработка фото производится следующим образом: после получения фото из функции «launchCamera» или «launchImageLibrary» запускается скрипт с нейронной сетью, в которую подается полученное фото. Фото обрабатывается и нейронная сеть выдает ответ в виде строки с названием породы собаки. До этого шага мы специально назвали поля id в массиве с породами так же, как классы в нейронной сети и таким образом нейронная сеть выдает строку, по которой с помощью навигации передается нужная страница с породой собаки.

ЗАКЛЮЧЕНИЕ

В ходе дипломного проекта было создано мобильное приложение на React Native Expo, работающее на основе нейронной сети по распознаванию породы собаки. Было использовано множество библиотек и фреймворков, таких как: PyTorch, Pandas, React Native, React Native ImagePicker, React Native Navigation. Активно использовались: среда разработки Google Colab, язык программирования Python и JavaScript, а так же с помощью инструмента Figma был разработан дизайн мобильного приложения. Данное мобильное приложение позволяет быстро распознать породу и получить основные справочные сведения по ней. Его главным преимуществом является скорость, проста использования, минимальное количество действий для получения нужной пользователю информации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Using React Native Camera in your app [Электронный ресурс]. — URL: <https://www.fullstacklabs.co/blog/react-native-camera> (Дата обращения 17.05.2022). Загл. с экрана. Яз. англ.
- 2 PyTorch documentation [Электронный ресурс]. — URL: <https://pytorch.org/tutorials/> (Дата обращения 21.04.2022). Загл. с экрана. Яз. англ.
- 3 Нейронные сети на Python [Электронный ресурс]. — URL: <https://highload.today/nejronnye-seti-na-python-kak-napisat-i-obuchit/> (Дата обращения 18.04.2022). Загл. с экрана. Яз. рус.
- 4 *Галушкин, А.* Нейронные сети: основы теории / А. Галушкин. — Горячая линия - Телеком, 2012.
- 5 *Скотт, А.* Разработка на JavaScript. Построение кроссплатформенных приложений с помощью GraphQL, React, React Native и Electron / А. Скотт. — Издательский дом: Питер, 2021.
- 6 React Native Camera [Электронный ресурс]. — URL: <https://www.npmjs.com/package/react-native-camera> (Дата обращения 13.04.2022). Загл. с экрана. Яз. англ.
- 7 Использование маршрутизации React Navigation в React Native [Электронный ресурс]. — URL: <https://www.digitalocean.com/community/tutorials/react-react-native-navigation-ru> (Дата обращения 11.04.2022). Загл. с экрана. Яз. рус.