УДК 004.934.1'1

НЕЙРОННАЯ СЕТЬ ДЛЯ РАСПОЗНАВАНИЯ ГОЛОСОВЫХ КОМАНД

Сакович В.А., студент гр.150701

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – доктор техн. наук

Аннотация. В работе рассмотрен вопрос разработки нейронной сети для распознавания голосовых команд. Обоснован выбор архитектуры нейронной сети. Описан алгоритм обучения и тестирования нейронной сети. Обучение, описание и тестирование нейронной сети выполнялось с помощью языка программирования Python и библиотеки PyTorch.

Ключевые слова. Сверточная нейронная сеть, Мелспектрограмма.

ВВЕДЕНИЕ

Современные системы голосового управления основаны на нейронных сетях – математических моделях, способных обучаться и распознавать сложные закономерности. Такие сети могут анализировать человеческую речь, преобразовывать её в текст, что позволяет управлять устройствами без физического взаимодействия с ним. Эта технология применяется в умных домах, автомобильных системах, промышленной автоматизации и медицинских устройствах, делая взаимодействие с техникой более удобным и доступным для большего количества людей.

В данной работе разрабатывалась нейронная сеть для распознавания набора из восьми голосовых команд: «yes», «no», «up» «down», «left», «right», «go», «stop». Этот набор является оптимальным так как он может быть полезен для различных вариантов использования в технических системах и для выполнения определенных действий, таких как подтверждение или отклонение действий, навигация и т.д.

Разрабатываемая сверточная нейронная сеть (СНС), представляет собой специализированную архитектуру для обработки мелспектрограмм голосовых команд. Модель состоит из четырех последовательных сверточных блоков, каждый из которых включает слой двумерной свертки, нормализацию по минипакетам, нелинейную активацию и операцию субдискретизации (англ. *max-pooling*).

ПОДГОТОВКА НАБОРА ДАННЫХ

В качестве опорного набора данных использовался набор данных Speech Command Dataset (SCD) [1]. В нем есть все перечисленные выше команды, которые представлены аудиозаписями длиной в одну секунду с частотой дискретизации 16 кГц.

Следующим шагом была подготовка класса тишины для того, чтобы нейронная сеть не делала ошибочных предсказаний в системе, которая будет ее использовать. Для этих целей был найден набор данных с записями окружающей среды [2]. Данные были преобразованы в такой же формат, как и команды из набора SCD. В результате был получен сбалансированный набор аудиоданных, в котором каждый класс представлен 2000 записей.

Подготовленная выборка разбивалась на два подмножества – обучающие данные и тестовые. Оптимальное разделение данных позволяет обеспечить обучение модели на достаточном объёме информации, при сохранении возможности её статистически значимой проверки на независимом наборе данных. Было выбрано соотношение обучающих данных к тестовым: 70/30. Во-первых, 70% данных хватает для качественного обучения модели, чтобы она могла выявить основные закономерности. Во-вторых, 30% тестовых данных – это достаточно большая выборка для надежной оценки ее реальной производительности. На диске набор данных хранился следующим образом: в каталоге train хранились папки с обучающими данными, а в каталоге test – с тестовыми данными. Для каждого класса в папке train было расположено 1400 аудиофайлов, а в папке с тестовыми данными было 600 файлов. Также был описан файл format.json для разделения папок с аудиоданными по классам. Файловая структура набора данных представленная на рисунке 1,а.

АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ

Для распознавания голосовых команд перед подачей на вход нейросети звуковые сигналы преобразуются в мелспектрограмму. Мелспектрограмма – это представление аудиосигнала в виде двумерного графика, где по оси абсцисс отложено время, по оси ординат — частота, представленная в шкале мелов, а по оси аппликат – энергия сигнала. Преимущество мелспектрограммы заключается в её способности выделять наиболее значимые для распознавания человеческой речи частотные диапазоны. Поэтому мелспектрограммы широко используются в задачах обработки звука с применением нейронных сетей [3]. На рисунке 1,б представлено изображение мелспектрограммы команды «down».

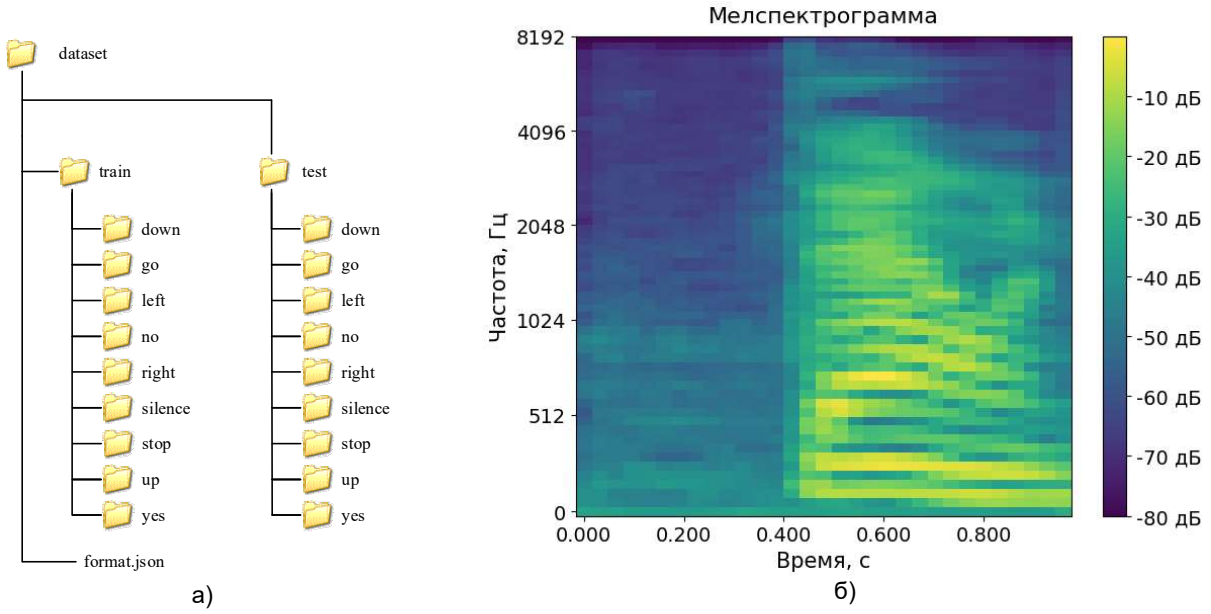

Рисунок 1 – Набор данных: а) файловая структура; б) пример мелспектрограммы команды «down»

Для обработки мелспектрограмм выбрана архитектура сверточной нейронной сети, так как она обладает высокой эффективностью при обработке изображений, а также хорошо показывает себя в обработке мелспектрограмм [4]. Архитектура СНС, на которую подается одна мелспектрограмма представлена на рисунке 2.

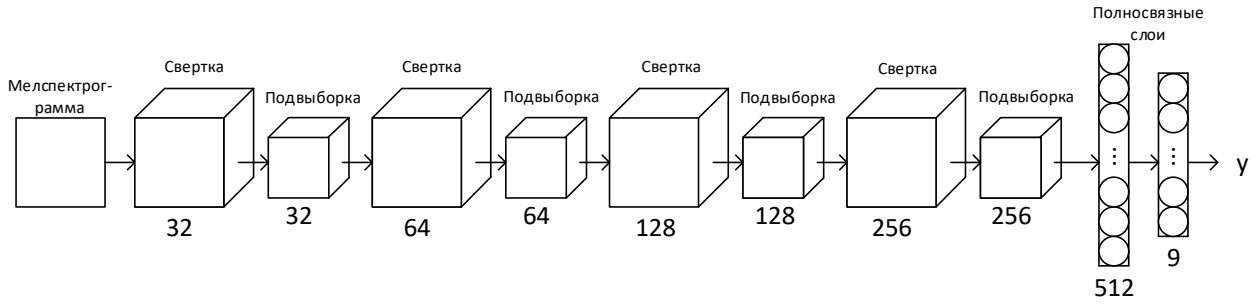

Рисунок 2 – Архитектура разработанной СНС

Разработанная СНС для обработки мелспектрограмм представляет собой модель, состоящую из четырех сверточных блоков, каждый из которых включает сверточный слой, пакетную нормализацию и операцию субдискретизации. Входом сети является минипакет из 32 спектрограмм размером 64×100 пикселей. Первый сверточный слой использует 32 фильтра размером 3×3 с сохранением исходных размеров. Последующие слои увеличивают количество фильтров до 64, 128 и 256 соответственно, сохраняя ту же структуру. После каждого сверточного слоя применяется операция субдискретизации с окном 2×2, которая уменьшает пространственные размеры вдвое, выделяя наиболее значимые признаки. После сверточных слоев данные преобразуются в одномерный вектор размером 6144 элемента (256×4×6) и проходят через два полносвязных слоя. Для предотвращения переобучения перед полносвязными слоями был добавлен слой дропаут, который во время обучения на каждом шаге случайным образом обнуляет часть нейронов в слое с вероятностью 50%. Далее идет первый полносвязный слой с 512 нейронами и активацией ReLU, который осуществляет нелинейное преобразование признаков, второй слой выдает итоговые предсказания для 9 классов.

ОБУЧЕНИЕ И ТЕСТИРОВАНИЕ МОДЕЛИ

После описания модели необходимо обучить модель. Для этого сначала была определена функция потерь – перекрестная энтропия:

$$E = -\sum_{n=1}^{N}\sum_{i=1}^{M} t_i^n \cdot \log y_i(x^n), \quad (1),$$

где N — количество примеров в выборке, M – количество классов, t_i^n – истинная метка для примера n и класса i, $y_i(x^n)$ — предсказанная моделью вероятность того, что пример x^n принадлежит классу i.

Эта функция потерь хорошо подходит для задач классификации [5]. Затем создается оптимизатор Adam, который будет обновлять параметры модели. Этот оптимизатор эффективно

адаптирует скорость обучения для каждого параметра модели индивидуально. Также был определен планировщик скорости обучения ReduceLROnPlateau, который автоматически уменьшает скорость обучения в 10 раз, если значение потерь не улучшается в течение 5 эпох. Перед началом обучения модель переносится на доступное вычислительное устройство.

Обучение проходит в цикле по эпохам. Было выбрано 10 эпох для обучения. В начале каждой эпохи модель переключается в режим обучения, что активирует такие слои как дропаут и нормализацию минипакетов.

Для каждой эпохи инициализируется переменная для накопления значений функции потерь, а также обнуляются градиенты оптимизатора, чтобы предотвратить их накопление. После выполняется прямой проход – данные пропускаются через модель, получаются предсказания, вычисляется значение функции потерь, сравнивающее предсказания с истинными метками. Далее выполняется обратное распространение ошибки, вычисляются градиенты, после чего оптимизатор обновляет параметры модели. После обработки всех минипакетов в эпохе планировщик обновляет скорость обучения на основе накопленных потерь. На рисунке 3,а представлен график ошибки в процессе обучения.

Следующим этапом после обучения модели является тестирование. Для этого модель была переведена в режим оценки, после этого в цикле обрабатываются тестовые данные без вычисления градиентов: для каждого минипакета получаются предсказания модели, определяется наиболее вероятный класс и сравнивается с правильными ответами. После обработки всех данных выводится точность модели, которая равна 95,5%.

Матрица ошибок для СНС на тестовом наборе представлена на рисунке 3,б. Полученные результаты свидетельствуют, что модель демонстрирует высокую эффективность в распознавании голосовых команд. Большинство предсказаний сосредоточено на главной диагонали матрицы, что указывает на правильную классификацию для подавляющего числа примеров. Особенно хорошо модель справляется с классами «right», «yes» и «silence». Тем не менее, матрица показывает и проблемные зоны. Несложно заметить, что модель допускала ошибки в тех командах, которые имели похожие звучания. Однако количество таких ошибок невелико и является некритичным. Высокая точность по всем классам подтверждает, что модель успешно обобщает признаки и хорошо справляется с основной задачей классификации голосовых команд.

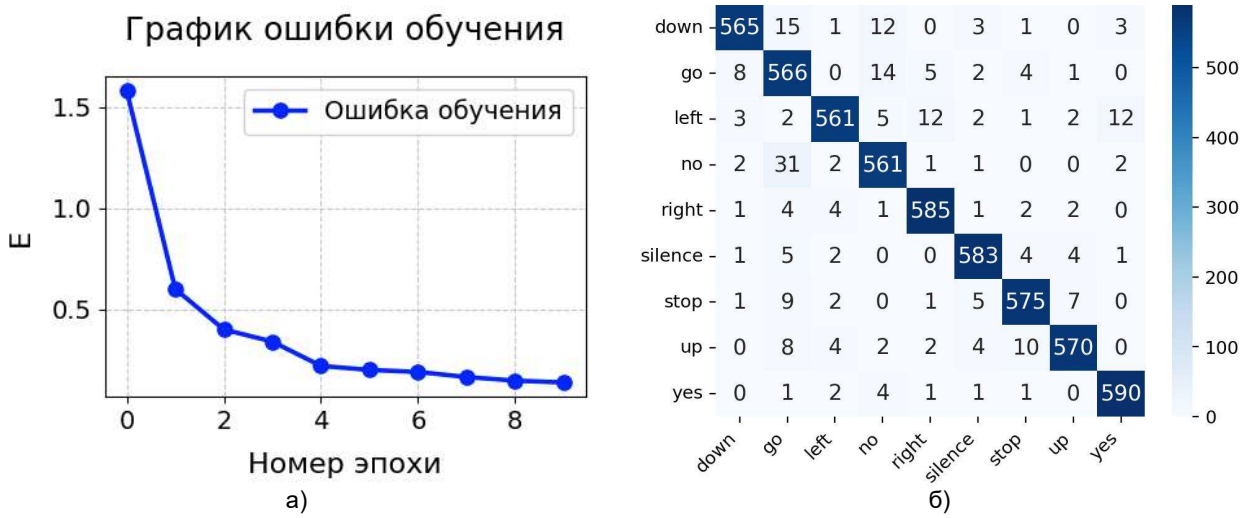

Рисунок 3 – Обучение модели: а) график ошибки обучения; б) Матрица ошибок

ВЫВОД

Полученная модель СНС продемонстрировала высокую эффективность, показав точность 95,5% на тестовых данных, что свидетельствует о ее способности успешно решать поставленную задачу распознавания голосовых команд. Еще одним преимуществом реализованной нейронной сети является ее архитектура – при достаточно высокой точности предсказаний модели, она является не слишком глубокой, что позволяет ее использовать во многих системах, которые не обладают высокими вычислительными мощностями.

Список использованных источников:

1. URL: https://tensorflow.google.cn/datasets/catalog/speech_commands. Дата доступа: 16.04.2025.

2. URL: https://github.com/karolpiczak/ESC-50/tree/master

3. Boyang Zhang Jared Leitner Sam Thornton Audio Recognition using Mel Spectrograms and Convolution Neural Networks Dept. of Electrical and Computer Engineering, University of California, San Diego, 2019 – 5 p.

4. Xuejiao Li, Zixuan Zhou Speech Command Recognition with Convolutional Neural Network, 2017 – 6 p.

5. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.: цв. ил.

UDC 004.934.1'1

NEURAL NETWORK FOR RECOGNIZING VOICE COMMANDS

Sakovich V.A.

Belarusian State University of Informatics and Radioelectronics[1], Minsk, Republic of Belarus

Vashkevich M.I. – Doctor of Science

Annotation. The paper considers the principle of operation of a neural network for recognizing voice commands. The choice of neural network architecture is justified. An algorithm for training and testing a neural network is described. The neural network was trained, described, and tested using the Python programming language and the PyTorch library.

Keywords. Convolutional neural network, Mel spectrogram.