РЕФЕРАТ

Выпускная квалификационная работа бакалавра содержит 70 страниц, 24 рисунка, 3 таблицы, 17 использованных источников.

МАШИННОЕ ОБУЧЕНИЕ, РАСПОЗНАВАНИЕ РЕЧИ, ЭЛЕКТРОМИОГРАФИЯ, КОННЕКЦИОНИСТСКАЯ ВРЕМЕННАЯ КЛАССИФИКАЦИЯ, МЕХАНИЗМ ВНИМАНИЯ, ТРАНСФОРМЕР, ЯЗЫКОВАЯ МОДЕЛЬ, НЕЙРОННЫЕ СЕТИ

В выпускной квалификационной работе бакалавра показано решение задачи распознавания тихой речи с использованием в качестве входных данных сигналов электромиографии речевых артикуляторов. В рамках этой задачи используются цифровые фильтры для предобработки входных данных. Архитектура модели машинного обучения базируется на алгоритме коннекционистской временной классификации и нейронной сети трансформер.

На основе предложенных методов была разработана и обучена модель машинного обучения, способная переводить тихую речь в текстовый формат. Полученные результаты демонстрируют перспективность разработанной системы для распознавания сигналов электромиографии.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей выпускной квалификационной работе применяют следующие обозначения и сокращения.

1. **фМРТ** – функциональная магнитно-резонансная томография.
2. **ЭМА** – электромагнитная артикулография.
3. **ЭКоГ** – электрокортикография.
4. **ЭМГ** – электромиография.
5. **ЭЭГ** – электроэнцефалография.
6. **BRNN** – bidirectional recurrent neural networks (двунаправленная рекуррентная нейронная сеть).
7. **CER** – character error rate (частота символьных ошибок).
8. **CNN** – convolutional neural network (сверточная нейронная сеть).
9. **CTC** – connectionist temporal classification (коннекционистская временная классификация).
10. **DenseNet** – densely connected convolutional networks (плотно связанные сверточные сети).
11. **FT** – fourier transform (преобразование Фурье).
12. **FR** – frequency ratio (отношение частот).
13. **GRU** – gated recurrent unit (закрытый рекуррентный модуль).
14. **LSTM** – long short-term memory (длительная кратковременная память).
15. **MAV** – mean absolute value (среднее абсолютное значение).
16. **MDF** – median frequency (медианная частота).
17. **MNF** – mean frequency (сумма интенсивности спектра).
18. **MPF** – mean power frequency (средний спектр мощности).

19. **NLP** – natural language processing (обработка естественного языка).
20. **PF** – peak frequency (пиковая частота).
21. **ReLU** – rectified linear unit (выпрямленный линейный блок).
22. **ResNet** – residual neural network (остаточная нейронная сеть).
23. **RIL** – relative information lost (относительная потеря информации).
24. **RMS** – root mean square (среднеквадратичное значение).
25. **RNN** – recurrent networks (рекуррентные нейронные сети).
26. **SSC** – slope sign change (изменение знака наклона).
27. **SSI** – simple square integral (простой квадратный интеграл).
28. **STFT** – short-time Fourier transform (кратковременное преобразование Фурье).
29. **TPS** – total power spectrum (общий спектр мощности).
30. **VAR** – variance (дисперсия).
31. **VCF** – variance of central frequency (отклонение центральной частоты).
32. **WER** – word error rate (частота ошибок в словах).
33. **WL** – wave length (длина волны).
34. **ZC** – zero crossing (пересечение нуля).

ВВЕДЕНИЕ

В современном мире, где технологии становятся все более важными в нашей жизни, появляются новые требования к удобству и эффективности управления устройствами. Одним из перспективных и инновационных подходов к решению этой задачи является создание систем безмолвного доступа, которые позволяют пользователю манипулировать устройствами только с помощью сигналов, генерируемых мышцами.

Электромиография (ЭМГ) – это метод изучения электрической активности мышц, который позволяет записать электрические сигналы, генерируемые мышцами при их сокращении. Применение технологий электромиографии для управления устройствами открывает широкие перспективы, особенно для людей с физическими ограничениями или в условиях, когда голосовое или визуальное управление невозможно или неудобно.

В большинстве случаев под распознаванием речи понимается преобразование аудио-последовательности записи голоса человека в текстовые данные. Однако, в некоторых случаях использование не только звуковой, но и другой информации позволяет улучшить модель или даже полностью заменить аудио-модель. Таким образом распознавание сигналов электромиографии для устройств безмолвного доступа можно рассматривать как задачу распознавания речи, но с уникальными способностями.

По аналогии с тем, как распознается устная речь человека, распознание сигналов электромиографии речевых артикуляторов требует анализа и интерпретации мимических мышечных движений, связанных с произношение звуков и слов.

Однако распознание сигналов речевых артикуляторов имеет свои уникальные аспекты. В отличие от анализа звуков речи, который основан на акустических сигналах, распознавание сигналов ЭМГ требует работы с электрическими сигналам, сгенерированными лицевыми и шейными мышцами. Это означает необходимость использования специализированных методов обработки и алгоритмов для анализа и интерпретации этих сигналов.

Целью данной дипломной работы является исследование и разработка методов распознавания сигналов электромиографии для возможности создания систем безмолвного доступа. Для достижения этой цели необходимо:

- провести обзор существующих подходов к обрабоке и анализу данных сигналов электромиографии;
- разработать алгоритма предобработки данных сигналов электромиографии на основе анализа исходного набора данных;
- исследовать методы обогащения признакового пространства для распрознавания тихой речи;
- провести обзор методов распознаванию речи;
- построить и обучить модель машинного обучения, способную переводить тихую речь в текстовый формат;
- провести анализ обученной модели машинного обучения.

Полученные результаты могут быть использованы не только для создания системы безмолвного доступа, способной взаимодействовать с различными устройствами, но и в областях медицины и реабилитации.

ОСНОВНАЯ ЧАСТЬ

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Вводные определения

В данной работе понятие ”тихая речь”означает способ произнесения слов, при котором сводится к минимуму проявление стандартных звуков. Это подразумевает, что артикуляторы перемещаются, но воздушный поток не проходит через них также интенсивно, как в обычной речи. Тихая речь, несмотря на свое название, не является полностью беззвучной из-за возможности издавать звуки при движении губ и языка, но она заметно более тихая по сравнению с обычным произношением.

Для более точного определения тихой речи, стоит провести сравнение с другими типами речи. Существуют разные спектры речи, определяемые своими признаками. Каждая категория включает в себя различные формы речевого проявления, и границы между ними иногда могут быть размытыми, но общими чертами спектра являются четыре различных уровня речевой активности: вокализованная речь, шепот, тихая речь и субвокальная речь.

Вокализация – это «нормальный» речевой режим, используемый для общения в повседневные ситуации. Источником звуков при вокализованной речи является как голос, вызванный активацией голосовых связок, так и другие ограничения потока воздуха в речевом тракте, такие как трение и взрывы.

Следующий режим является шепот. Он отличается от вокализованной речи отсутствием звонкости. При шепотной речи воздух проходит через голосовой тракт, но голосовые связки больше не активируются. Громкость шепотной речи может варьироваться в зависимости от того, с какой силой проталкивается воздух, но обычно она достаточно громкая, чтобы ее услышали другие люди, находящиеся поблизости. Самое тихое проявление шепота называется неслышимым шумом. Несмотря на название, неслышимый шум обычно имеет достаточный поток воздуха, чтобы произвести некоторый звук, но обычно он недостаточно громкий, чтобы его могли понять другие, и его необходимо улавливать с помощью специального стетоскопического микрофона. При неслышимом шуме поток воздуха очень слабый и может быть немного больше, чем поток, возникающий при нормальном дыхании.

Тихая речь — это когда поток воздуха уменьшается до такой степени, что сам поток воздуха не вызывает никакого звука, но речевые артикуляторы все еще двигаются, как при разговоре. Для произнесения тихой речи может потребоваться контроль дыхания, чтобы поддерживать поток воздуха ниже уровня, вызывающего звук.

Последний способ речи — это субвокальная речь, внутреннее появление слов в уме при чтении или мышлении. В субвокальной речи говорящий не делает сознательно никаких попыток пошевелить речевыми артикуляторами. Однако эта внутренняя речь часто сопровождается небольшой активацией речевых мышц, о которой говорящий не осознает. Иногда это включает в себя заметные движения губ, но может также варьироваться от более незаметной активации мышц, которую невозможно увидеть визуально.

1.2 Применение тихой речи

Существует множество различных приложений, в которых может быть полезна тихая речь. Здесь мы обсудим три широкие категории приложений: частное общение, общение с некоторыми формами ограниченной способности говорить и взаимодействие с устройствами [1].

Одна из основных областей применения безмолвной речи — это ввод данных для компьютеров, телефонов и других устройств. Речь может быть очень эффективным методом взаимодействия с устройствами, как за счет использования виртуальных помощников, так и для диктовки текста. Как правило, это намного быстрее, чем другие методы ввода слов, такие как клавиатура, и может быть особенно полезно для мобильных устройств, не имеющих полноразмерной клавиатуры. Однако пользователи могут счесть вокализованную речь неуместной в некоторых условиях из-за соображений конфиденциальности и желания не беспокоить других, и для облегчения этих проблем можно использовать тихую речь.

Еще одним потенциальным вариантом использования безмолвной речи может стать клиническое применение для людей, которые больше не способны произносить нормальную слышимую речь, но все еще используют большую часть своих речевых мышц. Например, это может быть полезно для пациентов, перенесших ларингэктомию, когда гортань была удалена из-за травмы или заболевания. Это также может быть полезно для пациентов с некоторыми типами

заболеваний, поражающих нервы или мышцы. Конечно, эффективность в этих случаях будет зависеть от того, насколько целы речевые мышцы и от наличия тренировочных данных.

В зависимости от различных вариантов использования есть два возможных результата, которые мы можем получить от системы бесшумной речи: первый из них это перевод в текстовый формат, второй же – это синтезирование звука. Однако озвучивание можно выполнить косвенно, сначала распознав текст, а затем за тем синтезировав звук с помощью преобразования текста в речь. Поэтому основной задачей будет распознавание текста.

1.3 Системы безмолвного доступа

Системы безмолвного доступа основаны на использовании сигналов, считываемых с датчиков не подверженных влиянию шума. К таким сенсорам могут относится датчики поверхностной электромиографии.

Исследования, относящиеся к системам безмолвного доступа, проводятся по всему миру. Их популярность обусловлена обширным применением во многих областях, начиная от медицины и заканчивая военной промышленностью.

На данный момент существует разработанная система, основанная на событийном потенциале P300. Данная система позволяет людям, страдающим от болезней, приводящих к параличу, общаться, считывая по одной букве в слове [2].

Подобные нейроинтерфейсы доказали свою эффективность, однако, к тому же они являются медленными, что делает их не пригодными к использованию в повседневной жизни. Также данные устройства требуют понимания их механизма работы, таким образом для их применения нужно достаточно долго учиться и тренироваться.

Как уже упоминалось ранее, системы безмолвного доступа могут применяться в военной индустрии. Данные устройства могут быть применены, когда сообщения необходимо доставить на достаточно большом расстоянии, но при этом другие виды коммуникации могут быть опасны или перехвачены.

Например, управление перспективных исследовательских проектов министерства обороны США в 2008 году выделило 4 миллиона долларов на разработку системы, позволяющей общаться на поле боя без использования вокализированной речи и основанной на использовании сигналов электроэнцефалограммы.

Системы безмолвного доступа могут применятся для широкого спектра задач. В зависимости от поставленной задачи используются необходимые входные данные, которые можно разделить на группы. Дальше рассматриваются наиболее популярные принципы управления системами безмолвного доступа.

1.3.1 Системы, основанные на распознавании жестов

Распознавание жестов — это процесс анализа и интерпретации движений человеческого тела с целью соотнесения конкретного жеста с желаемым результатом или командой. Жесты могут включать в себя движения рук, лица или других частей тела.

Для распознавания жестов существует несколько подходов. Один из них использует визуальные признаки, где камеры осуществляют отслеживание и анализ движений. Этот метод позволяет реализовать системы с аутентификацией, вводом информации и управлением, причем для работы такой системы достаточно простой видеокамеры. Однако такой подход имеет свои недостатки, такие как необходимость нахождения человека в кадре и потребность в качественном изображении, что может быть вызовом в некоторых условиях.

Другой подход к распознаванию жестов основан на сборе информации с датчиков, которые регистрируют физическое состояние человека [3]. Этот метод позволяет выделить более широкий набор характеристик для распознавания жестов. Однако он также влечет за собой определенные сложности, такие как выбор оптимального расположения датчиков на теле человека и создание алгоритмов для обработки шумных сигналов.

1.3.2 Системы, основанные на распознавании сигналов головного мозга

Рассмотрим системы, которые реагирует на мысленные намерения человека. Для реализации подобной системы зачастую считывают сигналы ЭЭГ, которые соответствуют воображаемой деятельности, которую мысленно совершает человек. В качестве примера можно привести исследования, которые направлены на анализ сигналов головного мозга для управления курсором мыши, т.е. движений влево или вправо.

Подобные системы довольно сложно реализовать, при этом для этого необходимо специальное дорогое оборудование, что делает его непригодным в повседневной жизни.

1.4 Элекромиография

ЭМГ сигнал, изображенный на рисунке 1.1, представляет собой электрический сигнал, который возникает в мышцах человека как в состоянии покоя, так и при их активации. Этот сигнал измеряется с помощью электромиографии, анализирующей электрическую активность мышц [4].

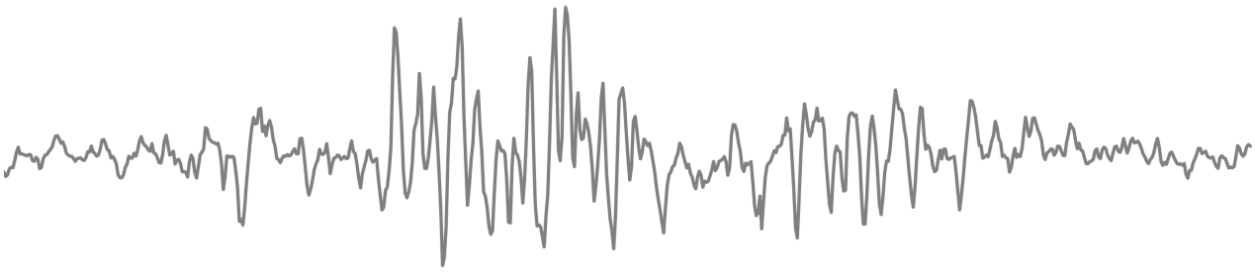


Рисунок 1.1 – Электромиографический сигнал

Когда мышцы активируются, происходит электрическая активация нейромышечных соединений. Сигнал данной активации показан на рисунке 1.2. Костный мозг генерирует электрические импульсы, которые приводят к сокращению мышцы. Перед выстрелом мышечные клетки имеют электрический потенциал на внешней мембране из-за дисбаланса заряженных ионов. Когда нервный сигнал заставляет мышцу активироваться, ионные каналы открываются, позволяя ионам проникнуть в клетку, что запускает химический процесс, за-

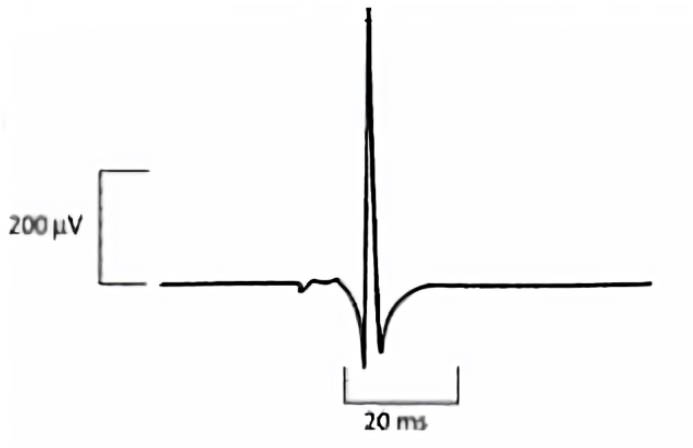


Рисунок 1.2 – Сигнал активации мышц

ставляющий мышцу сокращаться. После этого ионные насосы перемещают ионы обратно через клеточную мембрану. Это движение заряженных ионов в клетку и из нее вызывает распространение электрического импульса, и именно эти импульсы улавливают датчики ЭМГ.

1.5 Возможные альтернативы замены ЭМГ

Хотя основное внимание в этой работе будет уделено использованию ЭМГ в качестве входного сигнала для беззвучной речи, это не единственный возможный способ уловить речь, не зависящую от звука. Несколько альтернатив включают визуальные входные данные, электромагнитную артикулографию, ультразвук или сигналы мозга от ЭЭГ, ЭКоГ или фМРТ.

Один из альтернативных методов записи тихой речи — использование видео для визуального чтения по губам говорящего. Некоторыми преимуществами этого метода являются простота записи входных данных и большой объем доступных видеоданных речи, которые можно использовать для обучения. Одним из потенциальных недостатков является то, что видна только внешняя часть лица, а это может означать, что важная информация отсутствует. Есть также компромиссы с простотой использования, поскольку EMG может быть более удобным при ходьбе, а видео удобнее, когда вы сидите перед компьютером.

Еще одним возможным датчиком бесшумного речевого ввода является электромагнитная артикулография, или ЭМА, которая использует магниты, прикрепленные к губам и языку, для отслеживания их движений. Хотя ЭМА обладает очень точной информацией о движении речевых артикуляторов, что делает ее отлично подходит для использования в лаборатории, его необходимость прикреплять предметы к языку делает его слишком инвазионным для многих применений в качестве повседневного устройства связи.

Ультразвуковая визуализация внутренней части рта является еще одним возможным способом улавливания тихой речи. Преимущество ультразвука заключается в том, что он позволяет увидеть язык без установки датчиков во рту. Это может быть менее эффективно при захвате губ, поэтому иногда его комбинируют с визуальными данными для захвата этой информации. Еще одним недостатком является то, что современные ультразвуковые датчики зачастую более дорогие и громоздкие, чем ЭМГ или видео.

Наконец, есть несколько возможных входных данных, основанных на считывании сигналов мозга. Например, датчики ЭЭГ могут считывать электрические сигналы мозга с поверхности кожи точно так же, как датчики ЭМГ считывают мышцы. Однако из-за ослабления сигнала черепом эти датчики могут иметь слишком низкое разрешение, чтобы уловить достаточно информации для декодирования речи. Датчики ECoG, имплантированные внутрь черепа, могут собирать более детальную информацию, хотя для их имплантации требуется хирургическое вмешательство. Методы визуализации, такие как фМРТ, также могут использоваться для захвата речевой информации из мозга, но большой размер и стоимость этих аппаратов делают их непрактичными для многих случаев использования.

1.6 Методы регистрации ЭМГ

В настоящее время существует два вида методов регистрации миоэлектрической активности: инвазивные и неинвазивные. Инвазивный метод, известный как игольчатая электромиография, предполагает введение электрода непосредственно в исследуемую мышцу путем укола. Хотя этот метод обеспечивает наиболее точные результаты, он неудобен для повседневного использования,

особенно в области протезирования. Учитывая существенные недостатки инвазивных методов, неинвазивные или поверхностные методы стали более распространенными и предпочтительными.

Поверхностная ЭМГ – это метод записи миоэлектрических сигналов, который использует электроды, прикрепленные к поверхности тела человека. Этот метод более удобен, безболезненен, и позволяет использовать многоразовые электроды для регистрации мышечной активности. Поверхностная ЭМГ нашла широкое применение в медицине, биомеханике, изучении нервной деятельности, реабилитации и лечении двигательных расстройств.

1.7 Помехи сигналов ЭМГ

При получении биофизических сигналов, содержащих информацию, необходимую для конкретных измерений, дополнительно к основному сигналу также обрабатываются разнообразные шумы и помехи. Различные виды помех классифицируются по своему воздействию на сигнал, по происхождению, по вероятностным характеристикам и по энергетическому спектру.

Некоторые помехи, которые возникают в процессе регистрации сигналов, представляют собой искажения полезных данных, вызванные различными дестабилизирующими факторами, влияющими на измерения. Такие помехи могут быть вызваны, например, промышленным оборудованием или молнией. Помехи классифицируются по различным критериям, таким как их воздействие на сигнал, происхождение, вероятностные характеристики и энергетический спектр.

Источники таких помех подразделяются на внешние и внутренние. Внешние помехи обусловлены электромагнитными волнами, возникающими как в результате действий человека, так и имеющими природное происхождение. Сюда также относятся аппаратурные или инструментальные помехи, а помехами, вызванными деятельностью человека, могут быть, например помехи от переключателей, электродвигателей и других источников. Особое внимание уделяется сетевой помехе как приоритетному внешнему источнику помех.

Основные источники внутреннего шума включают в себя различные физиологические шумы внутри человеческого организма, а также шумы от электродов, связанные с работой других органов.

Типы помех включают в себя импульсные, флуктуационные и периодические помехи. Одной из разновидностей импульсных помех является шумовая помеха, проявляющаяся в виде отдельных импульсов (всплесков) или последовательности случайных импульсов. Источниками импульсных помех являются мгновенные всплески напряжения и тока в транспортных средствах, промышленном оборудовании и природных катаклизмах. Флуктуационные помехи характеризуются хаотическим процессом во времени, представленным непредсказуемыми случайными всплесками различной мощности. Они обычно имеют нормальное распределение с нулевым средним и оказывают существенное воздействие лишь на сигналы низкого уровня. Периодические помехи возникают из-за работы силовых электроустановок и линий электропередач, которые генерируют высокочастотные и низкочастотные поля.

Также искажения от артефактов движения существенно влияют на сигнал. Эти артефакты представляют собой резкие изменения напряжения, которые влияют на сигнал электромиографии (ЭМГ). Причиной появления таких изменений могут быть как движения электродов по поверхности кожи, так и изгибы кабеля, подключенного к электроду. Эти изменения обычно содержат максимальную энергию в диапазоне частот от 0 Гц до 2 Гц.

1.8 Распознавание сигналов ЭМГ

Подобно другим биомедицинским сигналам, главной целью исследования сигналов ЭМГ является извлечение информации о мышечной активности. Эта цель охватывает процесс распознавания паттернов сигналов ЭМГ. Следует отметить, что процесс распознавания паттернов сигналов ЭМГ обычно включает три этапа:

1) предварительная обработка сигнала, которая направлена на уменьшение воздействия внешних помех и улучшение отношения сигнал/шум;
2) извлечение полезной информации;
3) классификация.

1.9 Обработка входных данных

1.9.1 Разметка данных

Разметка данных играет ключевую роль в процессе первичной обработки информации. В контексте распознавания речи встает сложная проблема разметки речевых данных. Она заключается в том, что входные аудио данные представляют собой непрерывную последовательность звуков, в то время как текст представляет собой дискретную последовательность слов и символов.

Необходимость создания детальной разметки для каждой записи подразумевает определение временных меток для каждого звука или фонемы, что делает этот процесс сложным и затратным. Тем не менее, существуют специальные алгоритмы, такие как ”Connectionist Temporal Classification”(CTC), которые предназначены для решения данной проблемы.

Благодаря таким алгоритмам можно более эффективно устанавливать соответствие между записанными звуковыми сигналами и текстовой информацией, что снижает сложность и затраты на процесс разметки данных в задачах распознавания речи.

1.9.2 Фильтрация входных данных

Фильтрация биофизических сигналов, как, например, сигналов электромиографии, является важным этапом в обработке данных и требует специального внимания при разработке систем сбора и анализа биофизических сигналов.

Для получения достоверной информации от сигнала ЭМГ необходимо проводить фильтрацию сигнала с целью удаления шумов и помех. Шумы могут быть вызваны различными факторами, такими как электромагнитные помехи, движения пациента, смещение электродов и другие внешние воздействия в процессе сбора данных. Таким образом цель фильтрации - сохранить полезный биологический сигнал, удалив при этом нежелательные помехи, чтобы исследователи могли правильно интерпретировать данные с высокой точностью и достоверностью.

Фильтры можно классифицировать несколькими способами. Для обработки непрерывных во времени сигналов используются аналоговые фильтры, а дискретные во времени и квантовые по амплитуде сигналы обрабатываются цифровыми фильтрами.

Применение аналоговых фильтров происходит в электронике. Они производят антиалиасинговую обработку выводимого изображения, выборку радиостанций в радиоприемниках, разделение звукового сигнала на басы, твитер и др.

Цифровые же фильтры представлены в форме программного и аппаратного обеспечения вычислительных машин [5]. Они используются для спектрального анализа, обработки изображений, видео, речи и звука и др.

Цифровой фильтр можно рассматривать как функцию, которая может быть интерпретирована как во временной, так и в частотной областях. Во временной области она определяет ответ на единичный импульс от устройства или его математической модели, а в частотной области — влияние на амплитуду, фазу на определенной частоте. Переход между этими областями осуществляется с помощью преобразования Фурье, которое устанавливает однозначное соответствие между функциями в обеих областях.

Разделяют два типа цифровых фильтров: фильтры с конечной импульсной характеристикой (КИХ-фильтры) и фильтры с бесконечной импульсной характеристикой (БИХ-фильтры).

Импульсная характеристика — выходной сигнал динамической системы как реакция на входной сигнал в виде дельта функции Дирака, изображенной на рисунке 1.3, которая отображена на рисунке. В реальных физических системах входной сигнал представляет собой простой импульс минимальной ширины (равной периоду квантования для дискретных систем) и максимальной амплитуды.

КИХ и БИХ фильтры отличаются тем, что в КИХ-фильтрах выход зависит только от текущих и прошлых значений входных данных, в то время как в БИХ-фильтрах выход также зависит от предыдущих выходных значений сигнала, помимо входных значений.

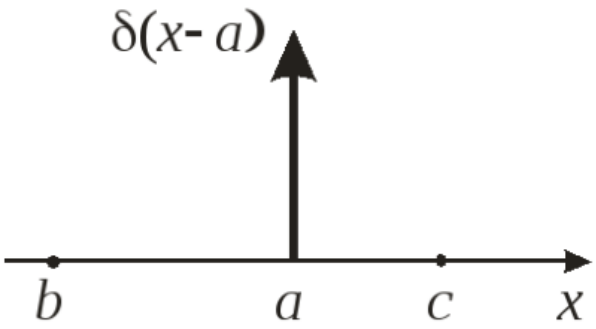


Рисунок 1.3 – Функция дирака

Фильтрация осуществляется с использованием различных методов цифровой обработки сигналов, таких как фильтры нижних и верхних частот, фильтры скользящего окна и др.

1.10 Выделение признаков

После проведения предварительной обработки электромиографического (ЭМГ) сигнала необходимо подвергнуть его процедуре извлечения полезной информации. Этот этап является критически важным в процессе разработки системы классификации паттернов. На данном этапе возможно снизить вычислительную сложность, улучшить актуальность обрабатываемых данных и избавиться от излишней информации. Основные особенности сигнала ЭМГ выделяют с помощью временных и частотных характеристик или сверточных нейронных сетей, которые мы рассмотрим подробнее. Важно отметить, что эта процедура играет ключевую роль в дальнейшем анализе и интерпретации сигнала, обеспечивая основу для выделения существенных паттернов и признаков для классификации.

1.10.1 Временные характеристики

Функции временной области вычисляются прямо из исходных данных и обладают низкой вычислительной сложностью. Результаты этих функций повышают точность классификации паттернов сигналов ЭМГ. Главным недостатком этих функций является их чувствительность к стационарности сигнала, что приводит к большим изменениям в работе с нестационарными сигналами.

Рассмотрим наиболее эффективные временные характеристики в распознавании сигналов ЭМГ.

1. Интегральная ЭМГ

   Для получения данной харкатеристики необходимо объеденить сигналы ЭМГ в определенном временном промежутке, размер такого промежутка определяется числом отсчетов N. Суммирование ЭМГ сигнала помогает в обнаружении точки активности мышц.

   Данная характеристика описывается следующей формулой:

$$\mathrm{G} = \sum_{i=1}^{N} \left| x_i \right|, \tag{1.1}$$

   где x_i – это измеренный сигнал,

   N – количество заданных временных промежутков.

2. Среднее абсолютное значение (MAV)

   Среднее абсолютное значение (MAV) очень схоже с предыдущей характеристикой, однако она менее подвержена воздействию шумов. MAV является одной из наиболее популярных характеристик при анализе сигналов ЭМГ. Данная характеристика необходима в вычислении степени сокращения мышц.

   Среднее абсолютное значение соответствует следующей формуле:

$$\mathrm{MAV} = \frac{1}{N} \sum_{i=1}^{N} \left| x_i \right|, \tag{1.2}$$

   где x_i – это измеренный сигнал,

   N – количество заданных временных промежутков.

3. Среднеквадратичное значение (RMS)

   RMS можно охарактеризовать как регистрацию постоянного мышечного услилия. Данная характеристика ищется по следующей формуле:

$$\mathrm{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2}, \tag{1.3}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков.

4. Длина волны (WL)

Длина волны (WL) – это мера комплексности сигнала, являющаяся постепенно накапливающейся суммой разностей каждого временного сегмента. Эта характеристика определяется формулой:

$$\mathrm{WL} = \sum_{i=1}^{N-1} \left| x_{i+1} - x_i \right|, \tag{1.4}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков.

5. Сумма квадратов (SSI)

Другое название данной функции – суммирование элементарных площадей. Данный параметр интерпретируется как мера энергии сигнала. Для вычисления этой характеристики суммируются квадраты значений амплитуды ЭМГ сигнала и вычисляется по формуле:

$$\mathrm{SSI} = \sum_{i}^{N} \left| x_i^2 \right|, \tag{1.5}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков.

6. Оценка дисперсии сигнала (VAR)

Дисперсия определяется как среднеквадратическое значение отклонения сигнала, так как среднее значение ЭМГ сигнала близко к нулю. Данная характеристика имеет смысл мощности сигнала. Таким образом дисперсия сигнала ЭМГ может вычислена следующим способом

$$\mathrm{VAR} = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \bar{x} \right)^2; \quad \bar{x} \approx 0, \tag{1.6}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков,

$\bar{x}$ – математическое ожидание (среднее значение).

7. Изменение знака наклона (SSC)

SSC – функция, которая позволяет определять положительное и отрицательное изменение знака наклона, рассчитывается в течение трех периодов.

$$\text{SSC} = \sum_{i}^{N} f\big[(x_i - x_{i-1}) \times (x_i - x_{i+1})\big], \tag{1.7}$$

$$f(x) = \begin{cases} 1, & \text{если } x \geq \Theta \\ 0, & \text{в остальных случаях} \end{cases}, \tag{1.8}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков,

Θ – пороговое значение.

8. Пересечение нулевой линии (ZC)

ZC – функция, отражающая сколько раз сигнал пересек нулевую линию с заданным пороговым значением. Данная характеристика вычисляется по формуле:

$$\text{ZC} = \sum_{i=1}^{N-1} \text{sgn}(x_i, x_{i+1}), \tag{1.9}$$

$$\text{sgn}(x_i, x_{i+1}) = \begin{cases} 1, & \text{если } x_i \times x_{i+1} < 0 \text{ и } x_i - x_{i+1} \geq \Theta \\ 0, & \text{в остальных случаях} \end{cases}, \tag{1.10}$$

где x_i – это измеренный сигнал,

N – количество заданных временных промежутков,

Θ – пороговое значение.

1.10.2 Частотные характеристики

Функции, которые рассматриваются в данном разделе, получаются с помощью применения преобразования Фурье и анализируются при помощи метода периодограммы. В отличие от функций во временной области, функции в частотной области отображают изменения сигнала в определенном диапазоне частот. Основные характеристики в частотной области включают спектральную плотность мощности, среднюю спектральную плотность мощности, медианную частоту и среднюю частоту [6].

Преобразование Фурье (FT) использует интегральное преобразование для представления исходного сигнала в виде базисных функций, выраженных через мнимые экспоненты с различными частотами, амплитудами и фазами. Это позволяет представить исходную функцию как сумму синусоид с различными характеристиками.

Однако преобразование Фурье не дает информации о том, как меняется частотный состав сигнала во времени. Для анализа этих изменений необходимо разбить сигнал на временные сегменты и вычислить спектр для каждого отрезка. Для этой цели используется кратковременное преобразование Фурье (STFT).

STFT определяется следующей формулой:

$$\mathrm{STFT}(k, l) = \sum_{n=0}^{N-1} h(n) x(n + lL) e^{-i\omega_k n}, \tag{1.11}$$

где $x(t)$ – входной сигнал,

$h(n)$ – оконная функция,

$k = \overline{0, N-1}$ – частотный индекс,

L – временной шаг анализа (расстояние между соседними фреймами),

l – номер фрейма анализа.

Периодограмма является функцией от частоты, используемой для оценки спектральной плотности. Её применение позволяет идентифицировать различные компоненты сигнала ЭМГ, включая постоянную, сезонную и случайную составляющие. Этот метод также способствует обобщенному анализу структуры временного ряда, который представляет собой последовательность данных, собранных с датчиков через определенные временные интервалы.

Рассмотрим наиболее эффективные частотные характеристики в распознавании сигналов ЭМГ [7].

1. Пиковая частота (PF)

   Пиковая частота – это частота, на которой достигается максимальная мощность в спектре. Данная характеристика определяется следующим образом:

$$\text{PF} = \max(P_j); \quad j = \overline{1, N}, \tag{1.12}$$

   где P_j – спектр мощности сигнала,
   N – количество заданных временных промежутков.

2. Средняя частота (MNF)

   Средняя частота рассчитывается как сумма произведений спектра мощности ЭМГ на частоту, разделенная на суммарную мощность спектра. MNF также называют центральной частотой или спектральным центром тяжести.

   Связь между частотным контентом и рекрутированием основана на том, что средняя частота спектра мощности отображает скорость распространения электрических нервных импульсов по мышечным волокнам.

   MPF имеет низкую эффективность в оценке скорости импульсов двигательных единиц, но при этом может использоваться для изучения качественного состава рекрутируемых волокон.

$$\text{MNF} = \frac{\sum\limits_{j=1}^{N} f_j P_j}{\sum\limits_{j=1}^{N} P_j}, \tag{1.13}$$

   где P_j – спектр мощности сигнала,
   f_j – частота сигнала,
   N – количество заданных временных промежутков.

3. Средний спектр мощности (MPF)

$$\text{MPF} = \sum_{j=1}^{N} \frac{P_j}{N}, \tag{1.14}$$

где P_j – спектр мощности сигнала,

N – количество заданных временных промежутков.

4. Общий спектр мощности (TPS)

Общий спектр мощности является совокупностью всех спектров мощности в исследуемом сигнале и является спектральным моментом нулевого порядка. TPS определяется по следующей формуле

$$\text{TPS} = \sum_{j=1}^{N} P_j = \text{SM}_0. \tag{1.15}$$

Спектральный момент (SM_i) порядка i создает новую функцию на основе спектра мощности и определяется как

$$\text{SM}_n = \sum_{j=1}^{N} P_j f_j^n; \quad n = \overline{1, n}, \tag{1.16}$$

где P_j – спектр мощности сигнала,

f_j – частота сигнала,

N – количество заданных временных промежутков.

5. Отношение частот (FR)

$$\text{FR} = \sum_{j=\text{LLC}}^{\text{ULC}} P_j - \sum_{j=\text{LHC}}^{\text{UHC}} P_j, \tag{1.17}$$

где P_j – спектр мощности сигнала,

ULC и LLC – верхняя и нижняя граница низкой частоты,

UHC и LHC – верхняя и нижняя границы высокой частоты.

6. Оценка дисперсии центральной частоты (VCF)

$$\text{VCF} = \frac{1}{\text{SM}_0}\sum_{j=1}^{N} P_j\left(f_j - f_c\right)^2 = \frac{\text{SM}_2}{\text{SM}_0} - \left(\frac{\text{SM}_1}{\text{SM}_0}\right)^2, \tag{1.18}$$

где P_j – спектр мощности сигнала,

f_j – частота сигнала,

N – количество заданных временных промежутков,

SM_i – спектральный момент.

7. Медианная частота (MDF)

Медианная частота – это разделение двух равных по амплитуде спектров.

$$\sum_{j=1}^{\text{MDF}} P_j = \sum_{j=\text{MDF}}^{N} = \frac{1}{2}\sum_{j=1}^{N} P_j, \tag{1.19}$$

где P_j – спектр мощности сигнала,

N – количество заданных временных промежутков.

1.10.3 Сверточные нейронные сети

Рассмотрим сверточные нейронные сети (CNN) для выделения признаков ЭМГ сигналов. Сети такого вида показали достаточно хорошие результаты при распознавании изображений. Такая особенность CNN позволяет рассматривать не каждый сигнал по отдельности, а все вместе как будто это изображение. В дальнейшем будем считать, что совокупность полученных от разных датчиков данных в одно и тоже время является изображением. Одним из важных преимуществ данной реализации заключается в устойчивости к изменениям масштаба и смещениям изображения.

Основное различие сверточных нейронных сетей от временных и частотных методов заключается в их способности автоматически и эффективно выделять ключевые признаки из изображений без необходимости ручного извлечения характеристик, что также приводит к их постоянному улучшению в процессе обучения.

Архитектура CNN базируется на сверточных слоях, ответственных за начальное извлечение признаков из входных данных с помощью фильтров - небольших матриц с весами. Применение фильтра к изображению основано на математической операции, известной как свертка.

На рисунке 1.4 показана работа свертки. Фильтр перемещается по всему изображению с определенным шагом (страйдом), умножая веса фильтра на соответствующие значения изображения на каждом этапе, за последующим суммированием. Результатом этой операции является формирование карты признаков, отображающей степень соответствия фильтра каждому локальному участку изображения. Для обеспечения анализа признаков по всему изображению, включая его края, и сохранения пространственного размера карты признаков часто используется паддинг - добавление пикселей вокруг начального изображения.

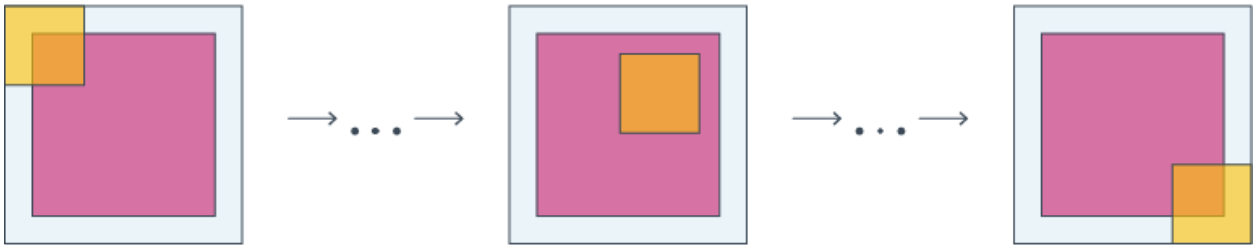


Рисунок 1.4 – Работа свертки

Представим данную операцию на математическом языке. Допустим входным изображением является матрица I, а фильтром – матрица F, то свертка в позиции (i, j) вычисляются следующим образом

$$C(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} I(i+u, j+v) \times F(u,v). \tag{1.20}$$

При этом картка признаков собирается из элементов $C(i, j)$.

При проектировании сверточных нейронных сетей возникает множество трудностей. Одной из таких является то, что в попытке улучшить качесто модели путем увеличения слоев в сети появляется проблема исчезающего градиента [8]. Она заключается в том, что по мере распространения градиента обратно к начальным слоям он может становиться чрезвычайно малым из-за многократ-

ного умножения. Это в свою очередь может привести к быстрому снижению эффективности. Для решения появившейся проблемы используют две популярные архитектуры сверточных нейронных сетей: ResNet и DenseNet. Рассмотрим их более подробно.

1.10.4 Архитектура ResNet

Residual Networks (ResNet) – это нейронная сеть, решающая проблему затухающего градиента. Благодаря этому стало возможно успешно обучать глубокие сети с более чем 150 соями. Данная архитектура была разработана исследователями Microsoft Research в 2015 году.

В ResNet была введена концепция, называемая остаточными боками. В ней используются пропускные соединения, которые соединяют слои с другими слоями, при этом пропускают некоторые слои между ними. Повторные соединения определяются путем объединения остаточных блоков вместе. Описанная архитектура изображена на рисунке 1.5.

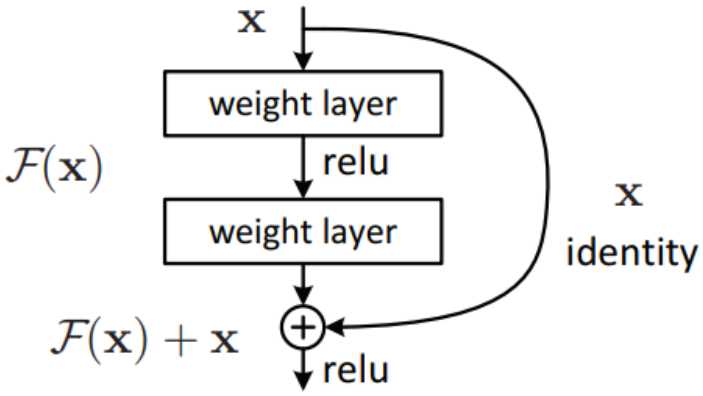
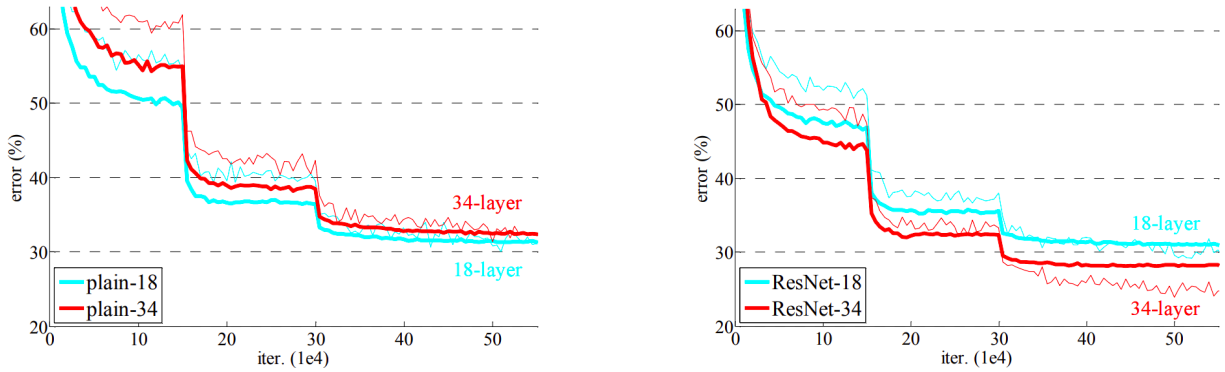


Рисунок 1.5 – Остаточный блок

Основная идея данного подхода заключается в том, что такое построение архитектуры позволяет пропускать уровни, ухудшающие эффективность модели. Именно поэтому получается обучать довольно глубокие нейронные сети, не обращая внимание на проблему затухающего градиента.

На рисунке 1.6 показаны результаты обучения 18-ти и 34-х уровневых сетей обычных сверточных нейронных сетей и архитектуры ResNet. ResNet показывает лучшие результаты при большей глубине сети.



(a) Обычная сверточная сеть (b) Архитектура ResNet

Рисунок 1.6 – Обучение сверточных сетей

1.10.5 Архитектура DenseNet

Плотно связанные сверточные сети (DenseNet) – это архитектура сверточной нейронной сети с прямой связью (CNN), которая связывает каждый уровень с любым другим уровнем. Это позволяет сети более эффективно обучаться за счет повторного использования функций, следовательно, уменьшая количество параметров и улучшая градиентный поток во время обучения.

Архитектура DenseNet, изображенная на рисунке 1.7, основана на простом и базовом принципе: путем объединения карт объектов всех предыдущих слоев плотный блок позволяет каждому слою получить доступ к объектам всех предыдущих уровней. В классических CNN каждый слой имеет доступ только к характеристикам слоя, непосредственно перед ним.

DenseNet состоит из переходных слоев и плотных блоков. Каждый сверточный слой внутри плотного блока связан с любым другим слоем внутри блока. Это достигается путем соединения выходных данных каждого слоя со входными данными следующего слоя, создавая «короткую» ссылку. Переходные слои минимизируют размер карт объектов в плотных блоках, что позволяет сети эффективно расти.

Данная нейронная сеть показывает очень хорошие результаты при обучении и является улучшенной версией ResNet. Однако для обучения DenseNet требуются огромные вычислительные затраты, из-за этого данную архитектуру нечасто используют.

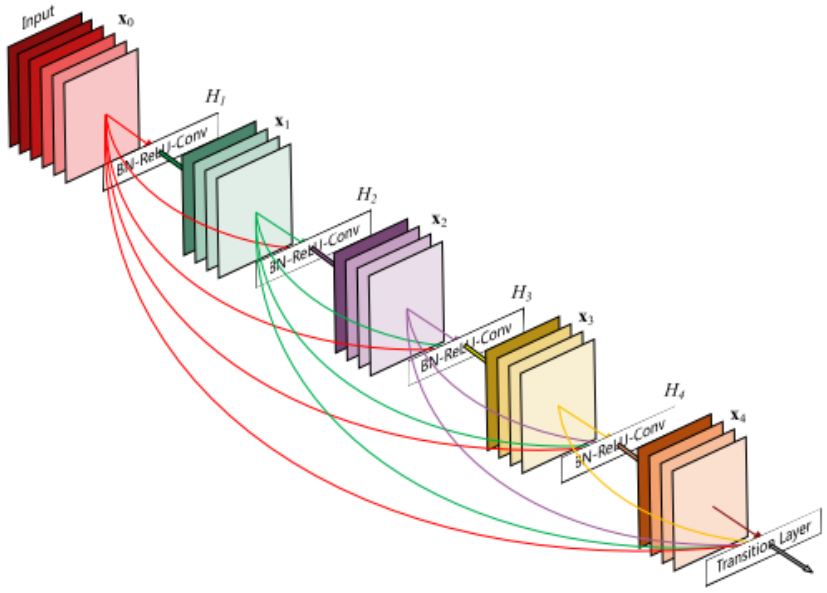


Рисунок 1.7 – Архитектура DenseNet

1.11 Классификация тихой речи

Распознавание речи с использованием ЭМГ имеет долгую историю исследований. Путем анализа и моделирования аудиосигналов удается преобразовать речевое содержание в текст для разнообразных приложений [9]. Тем не менее, возникают ситуации, когда звуковые сигналы нельзя получить четко.

Распознавание беззвучной речи устраняет ограничения автоматического распознавания звука, когда акустические сигналы не доступны. Однако распознавание тихой речи сложнее обычного.

Идея использовать ЭМГ для распознавания речи существует давно. Некоторые из первых исследований в области ЭМГ и речи были осуществлены Эдфельдтом в 1959 году при изучении субвокальной речи. В начале 2000-х годов люди начали применять системы автоматического распознавания речи, основанные на скрытых моделях Маркова, к речи на основе ЭМГ, включая тихую и субвокальную речь.

Алгоритмы классификации играют важную роль в распознавании беззвучной речи с использованием ЭМГ. В текущее время наиболее популярным подходом в распознавании речи является End2End.

1.11.1 End2end подход

С начала 2014 года увеличился интерес исследований к полному распознаванию речи. Ранее использовались традиционные методы, основанные на фонетике, требующие отдельных компонентов и обучения произношению, акустической и языковой модели [10]. Новые комплексные модели изучают все аспекты распознавания речи совместно, оптимизируя критерии для улучшения качества [11]. Это значительно упрощает процесс обучения и внедрения этих систем.

Проблема заключается в том, что речь — это непрерывная последовательность звуков, но на выходе требуется получить дискретную последовательность. До 2006 года не было эффективного способа моделировать этот процесс. Требовалась сложная разметка для каждой записи, указывающая момент произнесения каждого звука или буквы. Эта трудоемкая задача препятствовала проведению множества исследований в этой области. Однако в 2006 году статья Алекса Грейвса ”Connectionist temporal classification”(CTC) предложила решение данной проблемы, хотя вычислительных ресурсов не хватало на тот момент для его широкого применения. Реальные алгоритмы распознавания речи стали доступны значительно позже.

1.11.2 CTC алгоритм

Тринадцать лет назад был представлен алгоритм CTC как средство для обучения акустических моделей без необходимости сложной пофреймовой разметки, то есть без точного соответствия между фреймами входной и выходной последовательностей [12]. Следует отметить, что контекстно-зависимые фонемы, основанные на CTC, достигают высоких результатов в распознавании свободной речи.

Пофреймовое выравнивание подразумевает установление соответствия между фреймами сигнала и транскрипции, пример выравнивания показан на рисунке 1.8. Энкодер принимает акустические признаки и выдает скрытое состояние, на основе которого с помощью функции softmax вычисляются условные вероятности. Обычно энкодер представлен слоями LSTM или другими типами RNN, также возможно использование энкодера архитектуры transformer [13].

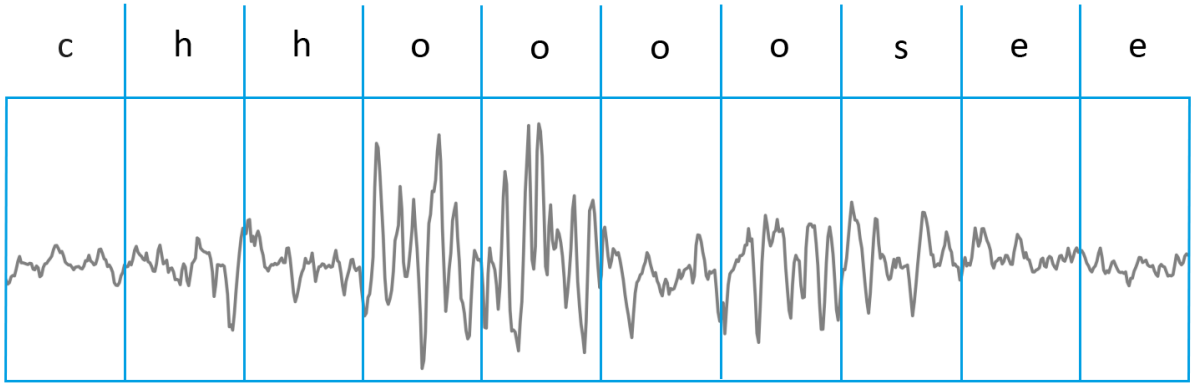


Рисунок 1.8 – пофреймовое выравнивание

Для выравнивания последовательностей в общий словарь вводится специальный токен («пробел»). Следовательно, вероятности символов, выводимые нейронной сетью, также включают вероятности наличия пустого символа для каждого временного шага.

Важно понимать отличие обычного пробела и специального пробела. Пробел – это реальный символ, который используется при написании текстов. Специальный пробел же означает отсутствие какого-либо символа и используется для обозначения границ между символами.

Основной задачей CTC алгоритма является максимизация вероятности предсказания правильной последовательности символов, благодаря обобщению возможных вариантов выравнивания.

CTC предлагает несколько преимуществ в задачах обучения от последовательности к последовательности, в том числе:

1) Упрощенный процесс обучения: CTC устраняет необходимость в явном согласовании на уровне фрейма входных данных и выходных последовательностей, делая процесс обучения более простым и эффективным;
2) Сквозное обучение: CTC обеспечивает сквозное обучение моделей, уменьшая необходимость в разработке сложных объектов и нескольких этапах обработки;
3) Гибкость: CTC может применяться к различным задачам последовательного обучения, таким как распознавание речи, текста и даже жестов.

1.11.3 Режим CTC loss

На вход алгоритма CTC поступает последовательность y. Данная последовательность представляет собой матрицу, в столбцах которой находятся вектора содержащие вероятности символов из алфавита A' в момент времени t. Тогда обозначим y_k^t как вероятность символа k в момент времени t. На рисунке 1.9 представлен пример матрицы вероятностной y.

Введем понятие пути в данной матрицы. Итак, путь (π) – это конечная последовательность, в которой каждый элемент соответствует символу из алфавита A'. Символы в данном последовательности закодированы номерами от 1 до n, из-за чего можно сделать обратное преобразование (декодирование) для получения последовательности символов.

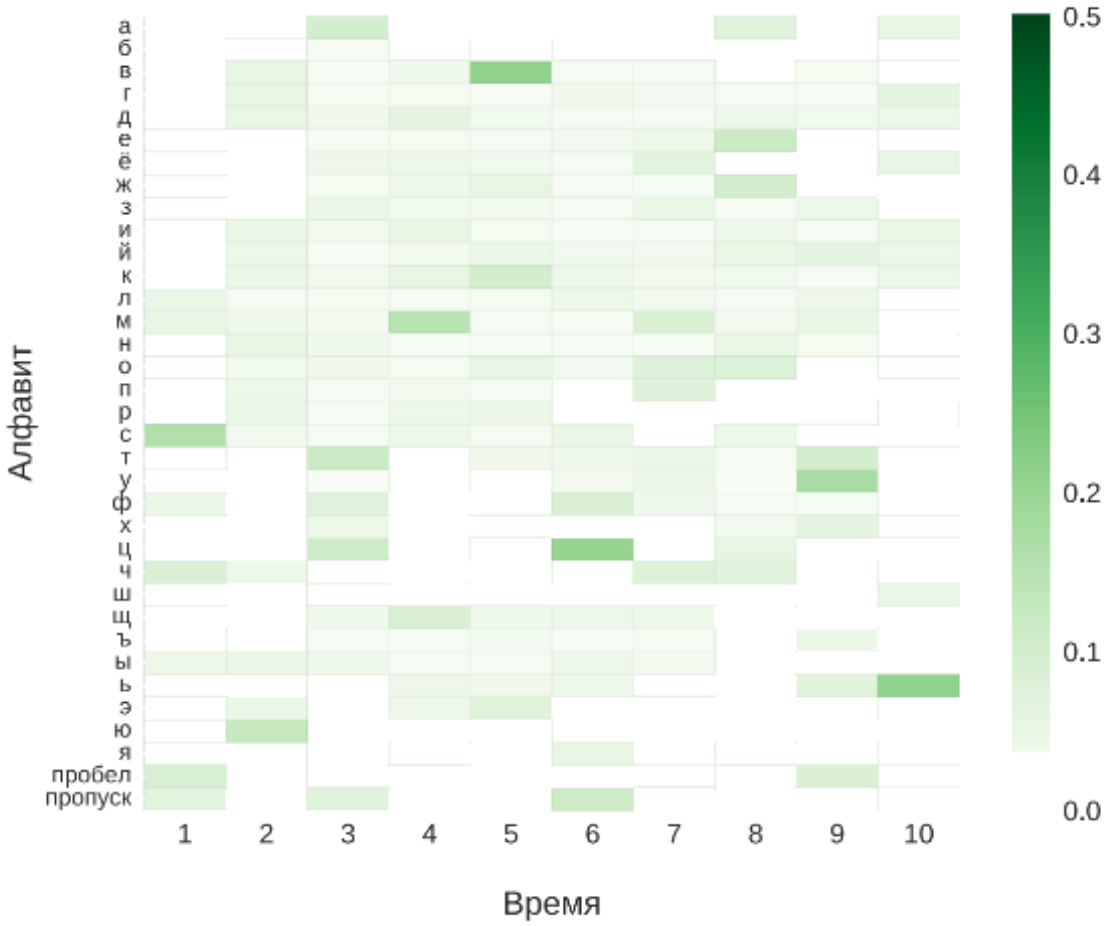


Рисунок 1.9 – матрица вероятностей

Рассмотрим различные пути в предсказанной трансформером матрице. С помощью следующей формуле можно вычислить вероятность каждого путя при заданном x:

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \quad \forall \pi \in {A'}^T \tag{1.21}$$

Распознавание речи основывается на том, что каждая произнесенная буква должна быть соотнесена с соответствующим символом, а полученные последовательности необходимо разделить одиночным пробелом. Однако записанные сигналы, соответствующие одному символу, могут быть произнесены на период времени, более долгий чем выбранный временной промежуток. Из-за чего в матрице предсказаний одному символу, ожидаемому в выходной последовательности, может соответствовать несколько столбцов.

Ключом к алгоритму CTC является использование специального токена, часто называемого пустым токеном. Это просто еще один токен, который модель будет предсказывать, и он является частью словаря.

Стоит отметить важную особенность алфавита. Для корректной работы алгоритма используется специальный символ, который также предсказывает нейронная сеть. Данный токен необходим для распознавания слов с повторяющимися буквами.

Для корректной интерпретации сигналов ЭМГ необходимо сделать ряд действий. Сначала необходимо соединить все повторяющиеся символы в пути, после чего следует удаление специальных токенов, входящих в алфавит A'.

Для обучения нейронной сети с помощью метода обратного распространения ошибки необходима функция, которая сможет вычислять ошибку предсказания для входной последовательности x и ожидаемого результата l. Помимо этого необходимо, чтобы эта функция была дифференцируема по переменным выходного слоя нейронной сети y_k^t. Определим такую функцию ошибки как минус натуральный логарифм от вероятности верного результата:

$$L_{\text{CTC}} = -\ln\big(p(l|x)\big) \tag{1.22}$$

Для получения вероятности верного результата l необходимо сложить вероятности всех путей:

$$p(l|x) = \sum_{\pi \in \mathbb{N}^T : \mathrm{B}(\pi) = l} p(\pi|x) \tag{1.23}$$

Так как нахождение всех путей, ведущих к результату l с помощью функции B перебором всех возможных путей очень затратно, необходим более оптимальный способ. Именно поэтому алгоритм CTC основан на алгоритме динамического программирования прямого-обратного хода.

Прежде чем рассматривать построение данного алгоритма, необходимо ввести некоторые обозначения. Формула $q_{1:p}$ означает, что из последовательности q выделили подпоследовательность, состоящую из первых ее p символов. Также, определим последовательность l', полученную из l путем вставки символа пропуска между любыми двумя символами l, а также в начале и в конце l.

На примере рисунка 1.10 разберем как строятся пути. Необходимо отметить, что для необходимо выполнение следующего условия

$$\mathrm{B}(\pi) = l \tag{1.24}$$

Для определения вероятностей всех путей, выделенных в таблице, которые начинаются с первой или второй строки первого столбца, а заканчиваются на пересечении строки s и столбца t, необходимо определить функцию $\alpha_t(s)$ следующим образом:

$$\alpha_t(s) = \sum_{\pi \in \mathbb{N}^T : \mathrm{D}(\pi_{1:t}) \in \left\{ l'_{1:s}, l'_{2:s} \right\}} \prod_{\tau=1}^{t} y_{\pi_\tau}^{\tau} \tag{1.25}$$

где D – преобразование, склеивающее одинаковые символы, стоящие последовательно, в один.

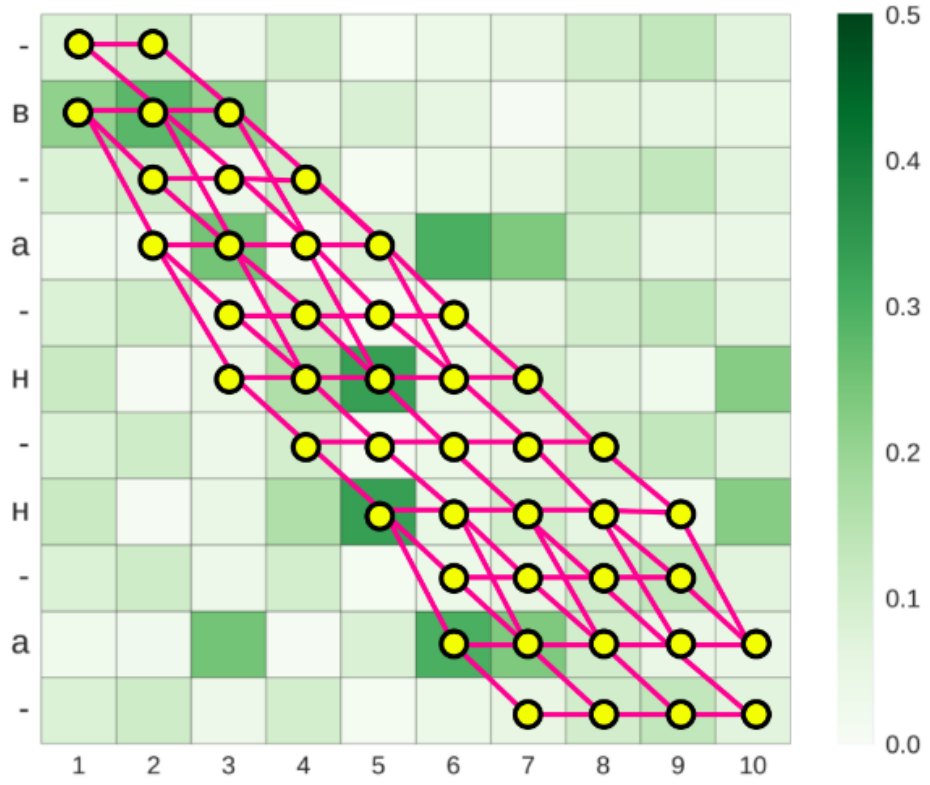


Рисунок 1.10 – построение путей

Для вычисления суммы вероятностей всех путей, с учетом того, что путь может заканчиваться либо специальным токеном, либо последним символом из l, нужно вычислить сумму функций альфа на последнем шаге времени от последнего символа l и от символа пропуска:

$$p(l|x) = \alpha_T(\left[l'\right] - 1) + \alpha_T(\left[l'\right]) \tag{1.26}$$

Теперь можно применять эту формулу для вычисления функции потерь CTC. Однако для обучения нейронной сети методом обратного распространения ошибки необходимо знать, как вычислять частную производную от функции L_{CTC} по переменным выходного слоя y_k^t:

$$\frac{\partial L_{\text{CTC}}}{\partial y_k^t} = -\frac{1}{p(l|x)} \frac{\partial p(l|x)}{\partial y_k^t}. \tag{1.27}$$

Для этого необходимо определить функцию, которая сможет вычислять вероятность всех путей, начинающихся в ячейке (s, t) таблицы и заканчивающихся на последнем символе или на символе пропуска, следующим образом:

$$\beta_t(s) = \sum_{\pi \in \mathbb{N}:\mathrm{D}(\pi_{1:T}) \in \left\{ l'_{s:[l']}, l'_{s:[l']-1} \right\}} \prod_{\tau=1}^{t} y_{\pi_\tau}^{\tau}. \tag{1.28}$$

Рассмотрим значения $\beta_t(s)$ для последнего столбца.

$$\beta_T([l']) = y_b^T \tag{1.29}$$

$$\beta_T([l'] - 1) = y^T \tag{1.30}$$

$$\beta_T(s) = 0, \forall < [l'] - 1 \tag{1.31}$$

Также как и с функцией $\alpha_t(s)$, для $\beta_t(s)$ существуют рекуррентные соотношения, вытекающие из определения функции B, позволяющие вычислить эту функцию для всех ячеек таблицы 2.3:

$$\beta_t(s) = \begin{cases} \big(\beta_{t+1}(s) + \beta_{t+1}(s+1)\big) y_{l'_s}^t, & \text{если } l'_s = b \text{ или } l'_{s+2} = l'_s \\ \big(\beta_{t+1}(s) + \beta_{t+1}(s+1) + \beta_{t+1}(s+2)\big) y_{l'_s}^t, & \text{иначе} \end{cases} \tag{1.32}$$

Заметим, что при данном l выражение $\alpha_t(s)\beta_t(s)$ в конкретной ячейке (s, t) дает суммарную вероятность всех путей, проходящих на шаге времени t через символ s:

$$\alpha_t(s)\beta_t(s) = \sum_{\pi \in \mathbb{N}^T:\mathrm{B}(\pi)=l, \pi_t = l'_s} y_{l'_s}^t \prod_{\tau=1}^{T} y_{\pi_\tau}^{\tau}. \tag{1.33}$$

Отсюда получаем что:

$$\frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t} = \sum_{\pi \in \mathbb{N}^T:\mathrm{B}(\pi)=l, \pi_t = l'_s} \prod_{\tau=1}^{T} y_{\pi_\tau}^{\tau} = \sum_{\pi \in \mathbb{N}:\mathrm{B}(\pi)=l, \pi_t = l'_s} p(\pi|x). \tag{1.34}$$

Тогда для каждого шага времени t мы можем вычислить

$$p(l|x) = \sum_{s=1}^{[l']} \frac{\alpha_t(s)\beta_t(s)}{y^t_{l'_s}}. \tag{1.35}$$

Возьмем частную производную от этого выражения по переменной выхода нейронной сети, соответствующей символу k на шаге времени t:

$$\frac{\partial p(l|x)}{\partial y^t_k} = -\frac{1}{\left(y^t_k\right)^2} \sum_{s:l'_s=k} \alpha_t(s)\beta_t(s). \tag{1.36}$$

И подставляя, получаем искомую частную производную для функции L_{CTC}:

$$\frac{\partial L_{\text{CTC}}}{\partial y^t_k} = \frac{1}{\alpha_T([l']-1) + \alpha_T([l'])} \frac{1}{\left(y^t_k\right)^2} \sum_{s:l'_s=k} \alpha_t(s)\beta_t(s). \tag{1.37}$$

1.11.4 Режим CTC decode

Для реализции режима CTC decode существует два алгоритма: максимальное декодирование (greedy search) и лучевой поиск (beam search). Рассмотрим их более подробно.

1.11.4.1 Алгоритм максимального декодирования

Самым вероятным результатом исходя из предсказанной матрицы, изображенной на рисунке 1.9, будет являться

$$h(x) = \underset{l \in A,\ l \leq T}{\arg\max}\, p(l|x) \tag{1.38}$$

Но так как число возможных результатов l для перебора слишком высоко мы будем использовать приближенные решения. Очевидным решением является выбрать путь собранный из элементов с максимальной вероятностью на каждом шаге t в матрице предсказаний:

$$h(x) \approx \text{B}(\underset{\pi \in \mathbb{N}^T}{\arg\max}\, p(\pi|x)) \tag{1.39}$$

Иллюстрацию этого решения можно увидеть на рисунке 1.11.

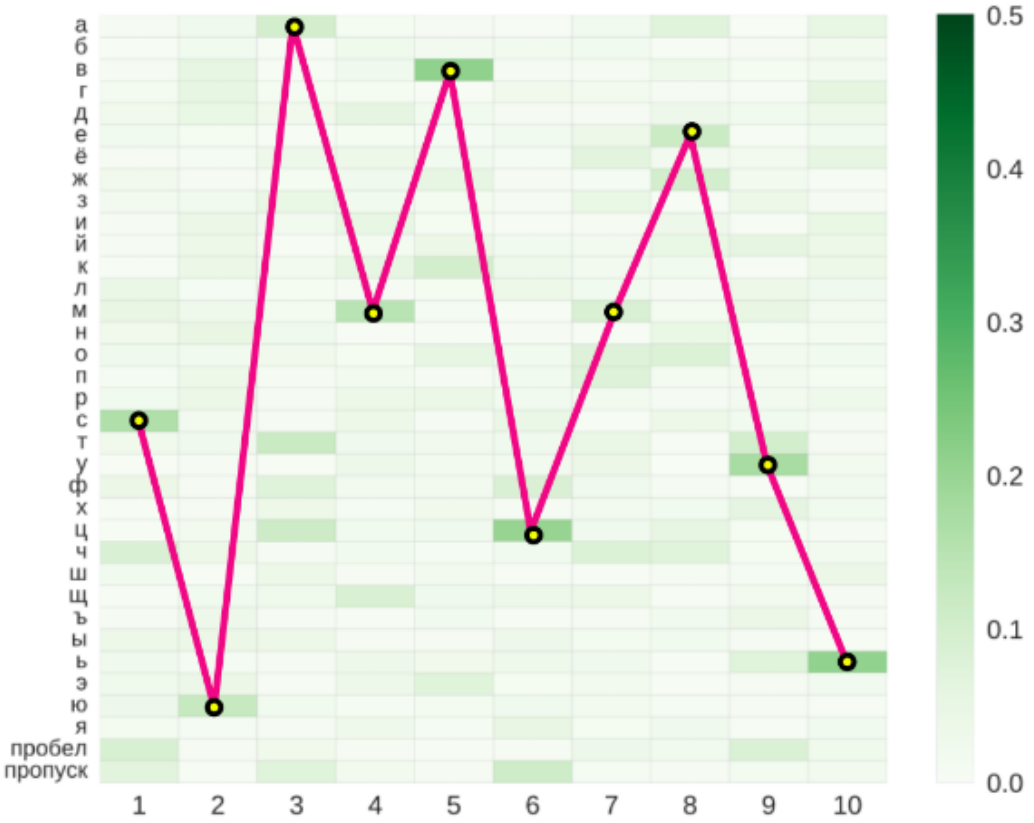


Рисунок 1.11 – решение алгоритмом greedy search

Но такое решение не всегда дает наиболее вероятный результат из-за того, что одному и тому же результату может соответствовать несколько путей и вероятность какого-то результата может оказаться больше, чем вероятность очевидного решения.

1.11.4.2 Алгоритм лучевого поиска

Декодирование поиска луча - это процесс, где алгоритм последовательно генерирует и оценивает различные куски данных (размером с луч), чтобы найти оптимальное решение. Основным параметром в этом методе является ширина луча, которая определяет количество символов, сохраняемых для анализа различных вариантов.

Рассмотрим данный алгоритм. На каждом шаге вычисляются две вероятности: $P_b(t, l)$ и $P_{nb}(t, l)$. P_b показывает вероятность того, что любой путь, который может быть преобразован операцией B в последовательность l, заканчивается специальным токеном. В свою очередь функция P_{nb} высчитывает вероятность того, что аналогичная последовательность заканчивается символом из алфавита A.

Также для использования алгоритма необходимо ввести пару вспомогательных функций: $\text{num}(c)$ и $\text{last}(s)$. Первая из них необходима для кодирования символа в его числовое значение, вторая же должна выдавать последний элемент конечной последовательности s.

Входные данные состоят из матрицы вероятностей, полученной из трансформера, обозначим элементы матрицы следующим образом

$$P(t, \text{num}(c)), \tag{1.40}$$

где t – номер столбца,

c – символ алфавита A'.

Также в данном алгоритме используется функция $P_{LM}(x)$, позволяющая вычислять вероятности последовательностей слов в речи.

Вначале алгоритма необходимо задать начальные значения (условия):

$$\begin{gathered} P_b(0, \varnothing) = 1, \\ P_n b(0, \varnothing) = 0, \\ R_0 = \varnothing, \end{gathered} \tag{1.41}$$

где R_t – множество лучших последовательностей на шаге t.

После инициализации начинается основная часть алгоритма, которая заключается в переборе символов c из алфавита A'. На каждом шаге алгоритма происходит следующая проверка:

Если c является специальным токеном, который обозначим символом «_», то вероятность $P_b(t, l)$ изменяется следующим образом:

$$P_b(t, l) \mathrel{+}= P(t, \text{num}(\_))\big(P_b(t-1, l) + P_{nb}(t-1, l)\big) \tag{1.42}$$

В противном случае возможны 3 ситуации:

1. $c = \text{last}(l)$, т.е. дублируется прошлый символ.
   Если символ является специальным токеном, тогда вероятность изменяется следующим образом

$$P_{nb}(t, l + c) \mathrel{+}= P(t, \text{num}(c))P_b(t - 1, l) \tag{1.43}$$

   иначе

$$P_{nb}(t, l) \mathrel{+}= P(t, \text{num}(c))P_{nb}(t - 1, l). \tag{1.44}$$

2. Новый символ является пробелом или достигнут конец временной последовательности.
   Происходит оценка текущей ситуации последовательности слов в последовательности с помощью языковой модели:

$$P_{nb}(t, l+c) \mathrel{+}= P(t, \text{num}(c))\big(P_{nb}(t-1, l)+P_b(t-1, l)\big)P_{LM}(l+c). \tag{1.45}$$

3. Если не выполнены условия предыдущих ситуаций.
   Задаем вероятность текущей последовательности с помощью формулы

$$P_{nb}(t, l + c) \mathrel{+}= P(t, \text{num}(c))\big(P_{nb}(t - 1, l) + P_b(t - 1, l)\big). \tag{1.46}$$

После того, как были вычислены все вероятности последовательностей, происходит их сортировка и выбор k последовательностей R_t с наибольшей вероятностью. В данном алгоритме число k является шириной луча.

После завершения всех итераций берется наиболее вероятная последовательность, являющаяся результатом декодирования. На рисунке 1.12 показаны первые шаги данного алгоритма.

1.11.5 Рекурентные нейронные сети

Рекуррентные нейронные сети, или RNN, это типы нейронных сетей, специализирующиеся на обработке последовательных данных, особенно в задачах обработки естественного языка.

Основное отличие RNN от традиционных нейронных сетей заключается в их способности использовать информацию из предыдущих временных шагов для воздействия на текущие входные и выходные данные.

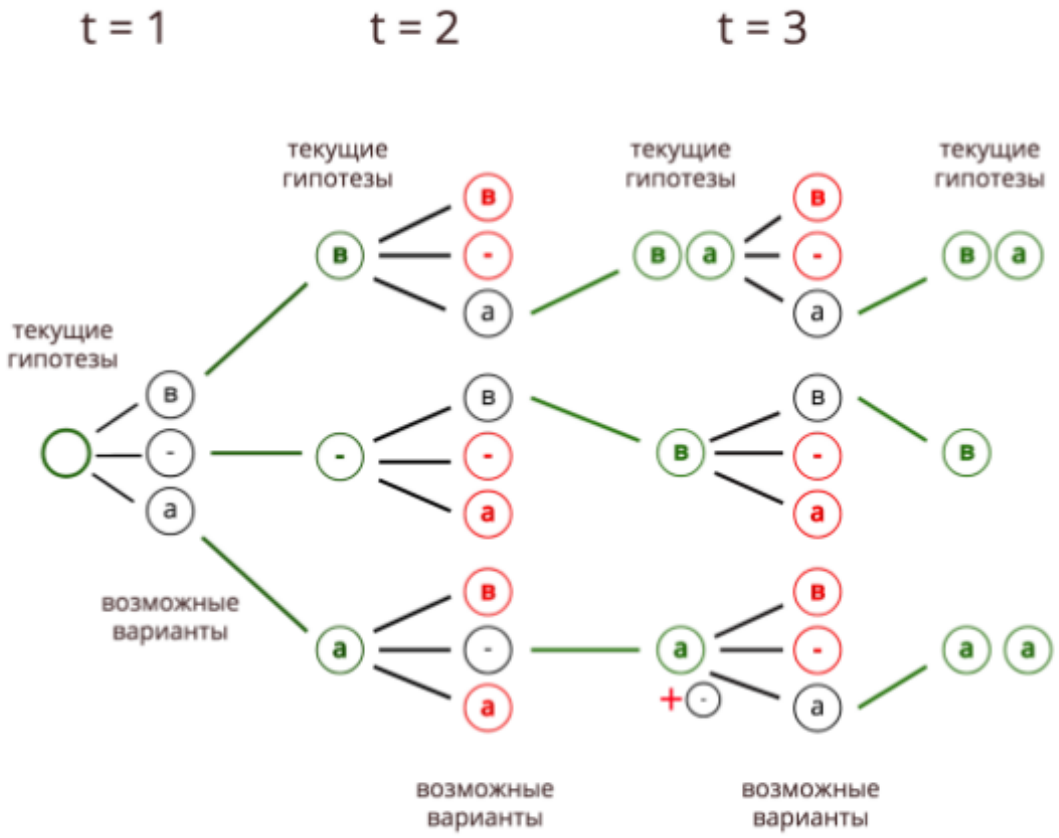


Рисунок 1.12 – пример работы алгоритма beam search

Еще одним важным аспектом RNN является совместное использование параметров на различных уровнях сети, в отличие от прямых нейронных сетей, где каждый узел имеет свои собственные веса.

Далее рассмотрим основные виды нейронных сетей.

1. Двунаправленные рекуррентные нейронные сети (BRNN)

   BRNN представляет собой отличный вид нейронной сети, основанный на архитектуре RNN. В отличие от однонаправленных RNN, которые опираются только на предыдущие входные данные для прогнозирования текущего состояния, двунаправленные RNN используют как предыдущие, так и будущие данные, чтобы повысить точность своих прогнозов.

2. Долговременная кратковременная память (LSTM)

   LSTM, представленные Зеппом Хохрайтером и Юргеном Шмидхубером, и являющиеся известной архитектурой RNN, были разработаны для решения проблемы исчезающего градиента [14]. Эти LSTM включают в себя специальные ”ячейки”в скрытых слоях нейронной сети, состоящие из трех

ключевых элементов: элемента ввода, элемента вывода и элемента забывания. Эти элементы контролируют поток информации, необходимый для точного прогнозирования выходных данных в сети.

3. Управляемые рекуррентные блоки (GRU)

   Этот вариант RNN напоминает LSTM в своем подходе к преодолению проблемы кратковременной памяти моделей RNN. Вместо использования информации о ”состоянии ячейки он оперирует с скрытыми состояниями и имеет два вентиля — вентиль сброса и вентиль обновления, вместо трех, как у LSTM. Аналогично шлюзам в LSTM, эти вентили определяют, какую информацию и в каком объеме сохранить.

1.11.6 Нейронная сеть transformer

Transformer – это нейронные сети, которые изучают контекст и понимание посредством последовательного анализа данных. Данная архитектура была представлена компанией Google за 2017 год как одна из самых передовых моделей, когда-либо разработанных. В моделях Transformer используется современный и развивающийся набор математических методов, обычно известный как внимание или самонаблюдение. Этот набор помогает определить, как удаленные элементы данных влияют друг на друга и зависят друг от друга.

Трансформаторы черпают вдохновение из архитектуры кодировщика декодера, используемой в RNNS, из-за их механизма внимания [15]. Transformer, в отличие от RNN, не выполняет обработку данных в последовательном порядке, что обеспечивает большее распараллеливание и более быстрое обучение.

Подход к обработке последовательностей целиком через внимание позволяет избавиться от такого понятия, как скрытое состояние, обновляющееся рекуррентно: каждый токен может напрямую «прочитать» любую часть последовательности, наиболее полезную для предсказания. В частности, отсутствие рекуррентности означает, что мы можем применять слой ко всей последовательности одновременно, так как матричные умножения прекрасно параллелятся.

1.11.7 Языковые модели

Языковая модель представляет собой вероятностное распределение по последовательностям слов. Она оценивает вероятность для каждой последовательности слов определенной длины m. Рассмотрим основыне типы языковых моделей.

1. Статистические языковые модели

   Статистические языковые модели – это тип моделей, которые используют статистические закономерности в данных для прогнозирования вероятности определенных последовательностей слов [16]. Основным подходом к построению вероятностной языковой модели является вычисление n - граммовых вероятностей.

   N-граммы представляют собой последовательность слов определенной длины, где число ”n”больше нуля. Для построения базовой вероятностной языковой модели вычисляются вероятности различных n-граммов (групп слов) в тексте путем подсчета их встречаемости и деления на количество появлений предыдущего слова. Этот подход основан на принципе Маркова, согласно которому вероятность следующего слова зависит только от предшествующего, игнорируя более дальний контекст. Хоть n-граммы просты и эффективны, они ограничены в учете долгосрочных зависимостей между словами в последовательности.

2. Нейронные языковые модели

   Нейронные языковые модели, использующие нейронные сети для прогнозирования последовательностей слов, обучаются на обширных текстовых наборах данных, позволяя им изучать основы языковой структуры. Эти модели способны работать с обширными словарями и обрабатывать редкие или неизвестные слова, используя векторные представления. В области обработки естественного языка (NLP) часто применяются архитектуры нейронных сетей, такие как рекуррентные нейронные сети (RNN) и трансформеры.

Нейронные языковые модели обладают более высокой способностью анализировать контекст по сравнению с традиционными статистическими моделями. Кроме того, они способны обрабатывать сложные языковые структуры и учитывать долгосрочные зависимости между словами.

1.11.8 Метрики оценки точности

Вопрос оценки эффективности систем распознавания речи является актуальным как для разработчиков собственных решений, так и для конечных пользователей. Первым шагом, который необходимо предпринять, - определить критерий оценки, то есть что именно будет измеряться, например, точность распознавания речи, скорость обработки, устойчивость к шуму в источнике данных. Затем следует выбрать показатель (метрику), который определит конкретное свойство в рамках выбранного критерия, например, процент правильно распознанных слов, время обработки аудиофрагмента, максимальный уровень шума и т.д.

Рассмотрим наиболее популярные метрики для оценки точности распознавания речи.

1. Word Error Rate (WER)

   Метрика WER представляет собой процент ошибок в словах, который измеряется по количеству операций замены, удаления и вставки слов при преобразовании одной строки в другую. Чем ближе значение метрики WER к нулю, тем выше точность распознавания, поскольку это указывает на меньшее количество ошибок в распознаваемых словах.

2. Relative Information Lost (RIL)

   Относительная потеря информации (RIL) — это показатель, который отражает степень информации, содержащейся в целевой фразе, и сохраняющейся в полученной моделью транскрипции. Он вычисляется как отношение условной энтропии целевой фразы к энтропии целевой фразы в полученном транскрипте.

3. Charater Error Rate (CER)

   CER – это метрика, аналогичная WER, но она измеряет процент ошибок не в словах, а в отдельных символах.

Имеются и другие редкие методы оценки качества, учитывающие различные типы операций с переменной значимостью. Например, замена оценивается более легко, чем вставка или удаление. Также имеются алгоритмы, учитывающие разнообразие слов и букв с различными весами, что применяется при оценке эффективности систем автоматического перевода.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Сбор данных

В данной работе входные данные, которые используются для распознания тихой речи, поступают с помощью поверхностной электромиографии или ЭМГ.

Поверхностная ЭМГ использует электроды, помещаемые на верхнюю часть кожи, для измерения электрических потенциалов, вызванных активностью близлежащих мышц. Размещая электроды вокруг лица и шеи, мы можем улавливать сигналы от мышц, важных для речи, которые могут включать челюсть, язык, губы, гортань и мягкое небо. Места размещения электродов описаны в таблице 2.1, а визуально их можно увидеть на рисунке 2.1. Эти места были выбраны так, чтобы располагаться близко ко многим основным речевым артикуляторам и иметь некоторое сходство.

Таблица 2.1 – Расположение датчиков

№	Местоположение
1	левая щека чуть выше рта
2	левая часть подбородка
3	ниже подбородка на 3 см
4	горло на 3 см левее кадыка
5	средняя часть челюсти справа
6	правая щека чуть ниже рта
7	правая щека на расстоянии 2 см от носа
8	задняя сторона правой щеки

Используемый набор данных содержит почти 20 часов сигналов ЭМГ лица от одного датчика, записанных со скоростью 1000 выборок в секунду. Сигналы ЭМГ, могут иметь величину до нескольких милливольт, но интенсивность может быть намного ниже в зависимости от силы мышечной активации и расстояния от электродов.

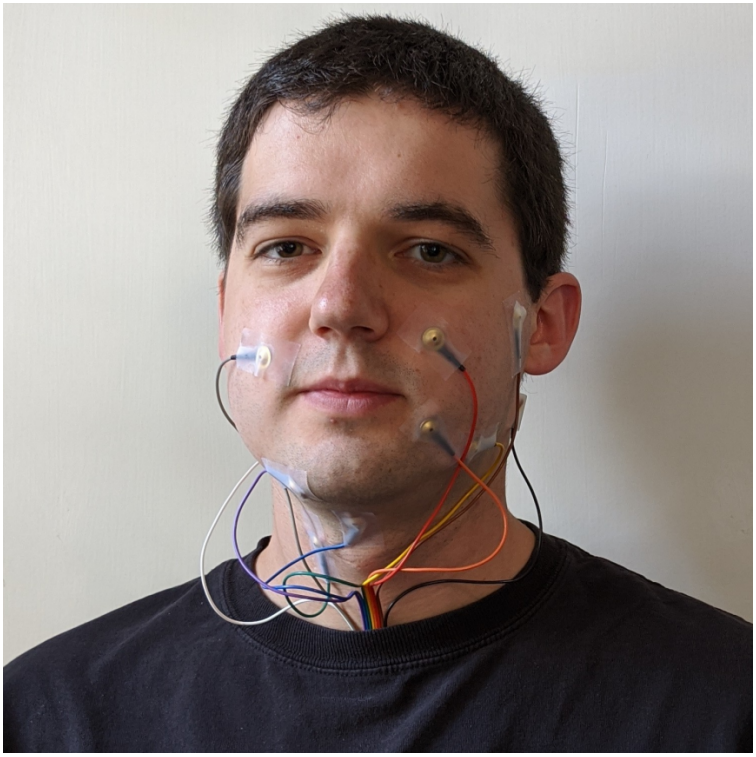


Рисунок 2.1 – Расположение датчиков на лице

2.2 Алгоритм предобработки данных

Прежде чем приступать к распознанию тихой речи, необходимо сделать достаточно важное мероприятие, которое заключается в предварительной обработке входных данных. В ходе обработки требуется очистить считываемый сигнал X от шумов и посторонних сигналов. Для этого в этой работе используются цифровые фильтры.

Цифровая обработка будет состоять из нескольких фильтров для уменьшения шума и нормализации сигналов. Во-первых, серия режекторных БИХ-фильтров Φ_1 с частотой, кратной 60 Гц, используется для уменьшения шума от сети переменного тока, который значительно проявляется в сигналах перед фильтрацией, часто с величинами, превышающими значения сигналов, которые мы стремимся захватить. Затем используется фильтр высоких частот Баттерворта Φ_2 с частотой среза 2 Гц для удаления постоянного смещения и медленного дрейфа из собранных сигналов, поскольку смещения могут варьироваться в широких пределах и обычно не содержат полезной информации. Для обоих фильтров используется фильтрация вперед-назад, чтобы избежать фазовых сдвигов. Реализованный алгоритм изображен на рисунке 2.2.

Таким образом был определен алгоритм P, который проводит предварительную обработку данных ЭМГ сигналов:

$$P(x) = \Phi_2(\Phi_1(x)) = \hat{x}, \quad x \in X \text{ и } \hat{x} \in \hat{X}, \tag{2.1}$$

где x – считанный сигнал ЭМГ,

$\hat{x}$ – результат предварительной обработки входных данных.

```
    def remove_drift(signal, fs):
        b, a = scipy.signal.butter(3, 2, "highpass", fs=fs)
        return scipy.signal.filtfilt(b, a, signal)

    def notch(signal, freq, sample_frequency):
        b, a = scipy.signal.iirnotch(freq, 30, sample_frequency)
        return scipy.signal.filtfilt(b, a, signal)

    def notch_harmonics(signal, freq, sample_frequency):
        for harmonic in range(1, 8):
            signal = notch(signal, freq * harmonic, sample_frequency)
        return signal

    def apply_to_all(function, signal_array, *args, **kwargs):
        results = []
        for i in range(signal_array.shape[1]):
            results.append(function(signal_array[:, i], *args,
            ↪  **kwargs))
        return np.stack(results, 1)
```

Рисунок 2.2 – Функции предобработки данных

2.3 Архитектура построенной модели

Разработанная архитектура в данной работе показана на рисунке 2.3. Она основана на современном подходе, который состоит в использовании совокупности CTC алгоритма и энкодера трансформера. Данную модель можно разделить на две части: обучаемую и алгоритмизированную.

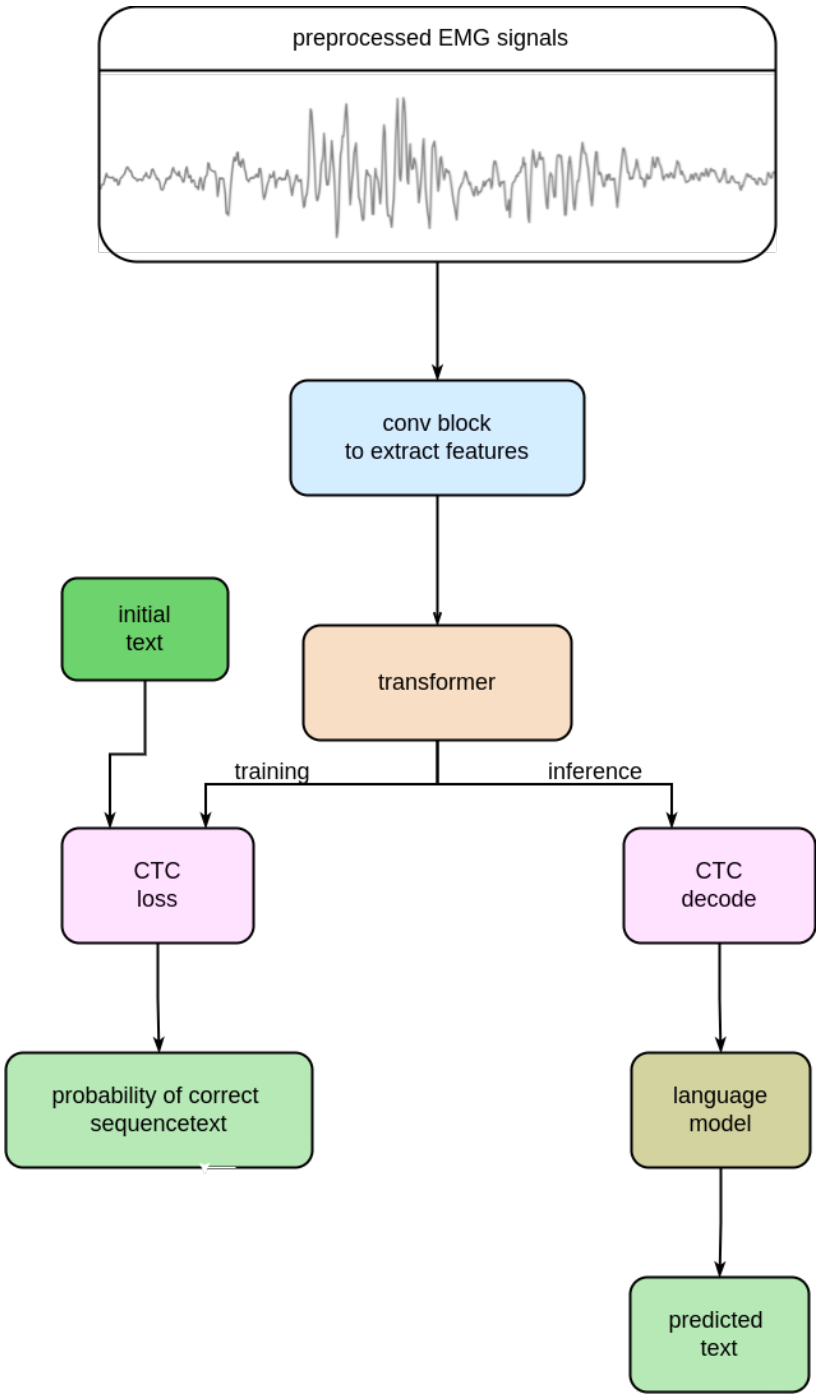


Рисунок 2.3 – Архитектура модели распознавания тихой речи

Обучаемая часть состоит из сверточной нейронной сети и трансформера. Этому блоку соответствует листинг кода, показанный на рисунке 2.4. Сверточная нейронная сеть нужна для выделения признаков. На ее вход поступает совокупность временных рядов, которые она преобразует в признаки. После этого полученные признаки поступают на вход к энкодеру трансформера, который получает из них матрицу вероятностей.

Следующая часть, называемая алгоритмизированной, состоит из алгоритма CTC. Данный алгоритм может работать в двух режимах: вычисления ошибки CTC и получении наиболее вероятной последовательности.

```
class Model(nn.Module):
    def __init__(self, num_outs):
        super().__init__()
        self.conv_blocks = nn.Sequential(
            ResBlock(8, FLAGS.model_size, 2),
            ResBlock(FLAGS.model_size, FLAGS.model_size, 2),
            ResBlock(FLAGS.model_size, FLAGS.model_size, 2),
        )
        self.w_raw_in = nn.Linear(FLAGS.model_size, FLAGS.model_size)
        encoder_layer = TransformerEncoderLayer(
            d_model=FLAGS.model_size,
            nhead=8,
            relative_positional=True,
            relative_positional_distance=100,
            dim_feedforward=3072,
            dropout=FLAGS.dropout,
        )
        self.transformer = nn.TransformerEncoder(encoder_layer,
        ↪  FLAGS.num_layers)
        self.w_out = nn.Linear(FLAGS.model_size, num_outs)

    def forward(self, x_raw):
        x_raw = x_raw.transpose(1, 2)
        x_raw = self.conv_blocks(x_raw)
        x_raw = x_raw.transpose(1, 2)
        x_raw = self.w_raw_in(x_raw)
        x = x_raw
        x = x.transpose(0, 1)
        x = self.transformer(x)
        x = x.transpose(0, 1)
        return self.w_out(x)
```

Рисунок 2.4 – Класс, реализующий архитектуру модели

2.4 Алгоритм выделения признаков

Первый шаг разработанной модели — извлечение признаков из сигналов ЭМГ с минимальной предварительной обработкой. Для этого добавляется набор слоев сверточной нейронной сети в начало модели преобразования, которые будут действовать как экстракторы функций. Эти слои обучаются вместе с остальной частью модели преобразования и дают модели возможность изучать свои собственные признаки ЭМГ сигналов.

Разработанная сверточная архитектура использует стек из трех остаточных блоков свертки, основанный на ResNet, но модифицированный для использования одномерных сверток. Архитектура, используемая для каждого блока свертки, показана на рисунке 2.5.

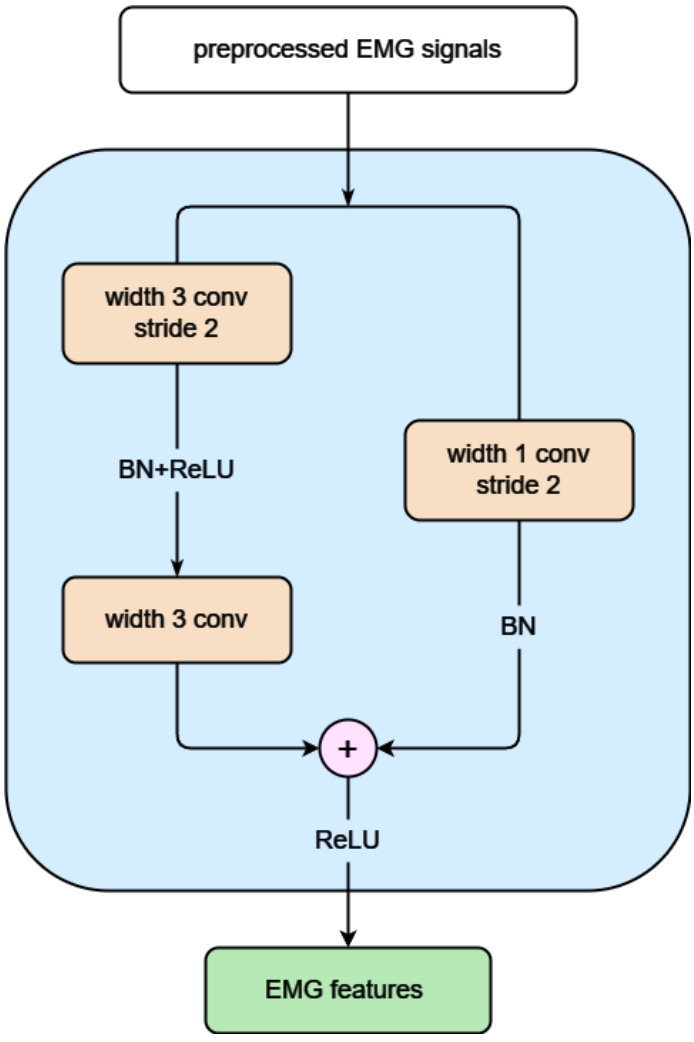


Рисунок 2.5 – Архитектура сверточного блока

Вдоль основного пути вычислений используются два слоя свертки шириной 3 по последовательности с выпрямленной линейной активацией (ReLU) между ними, а по другому «короткому» пути — одна свертка шириной 1 (линейное преобразование без последовательности) агрегирование. Конечный результат блока представляет собой сумму двух выходных данных пути, за которыми следует активация ReLU.

За каждой сверткой следует операция пакетной нормализации (на рисунке обозначается как БН). Шаги в начале блока установлены равными 2, так что каждый блок сокращается на 2, что дает общее уменьшение длины на 8 по трем слоям. Все свертки имеют размерность канала 768. Этому блоку соответствует листинг кода, показанный на рисунке 2.6.

```
class ResBlock(nn.Module):
    def __init__(self, num_ins, num_outs, stride=1):
        super().__init__()
        self.conv1 = nn.Conv1d(num_ins, num_outs, 3, padding=1,
        ↪  stride=stride)
        self.bn1 = nn.BatchNorm1d(num_outs)
        self.conv2 = nn.Conv1d(num_outs, num_outs, 3, padding=1)
        self.bn2 = nn.BatchNorm1d(num_outs)
        if stride != 1 or num_ins != num_outs:
            self.residual_path = nn.Conv1d(num_ins, num_outs, 1,
            ↪  stride=stride)
            self.res_norm = nn.BatchNorm1d(num_outs)
        else:
            self.residual_path = None

    def forward(self, x):
        input_value = x
        x = F.relu(self.bn1(self.conv1(x)))
        x = self.bn2(self.conv2(x))
        if self.residual_path is not None:
            res = self.res_norm(self.residual_path(input_value))
        else:
            res = input_value
        return F.relu(x + res)
```

Рисунок 2.6 – Класс, реализующий сверточный блок

2.5 Transformer

В теоретической части была дана общая информация о трансформере. В данном же разделе будет описана подробная информация о работе данного типа нейронной сети.

Классический трансформер состоит из двух основных частей, называемых энкодером и декодером [17]. Однако в работе будет использоваться только первая часть данной архитектуры, которая изображена в левой части рисунка 2.7, а реализация на рисунке 2.8.

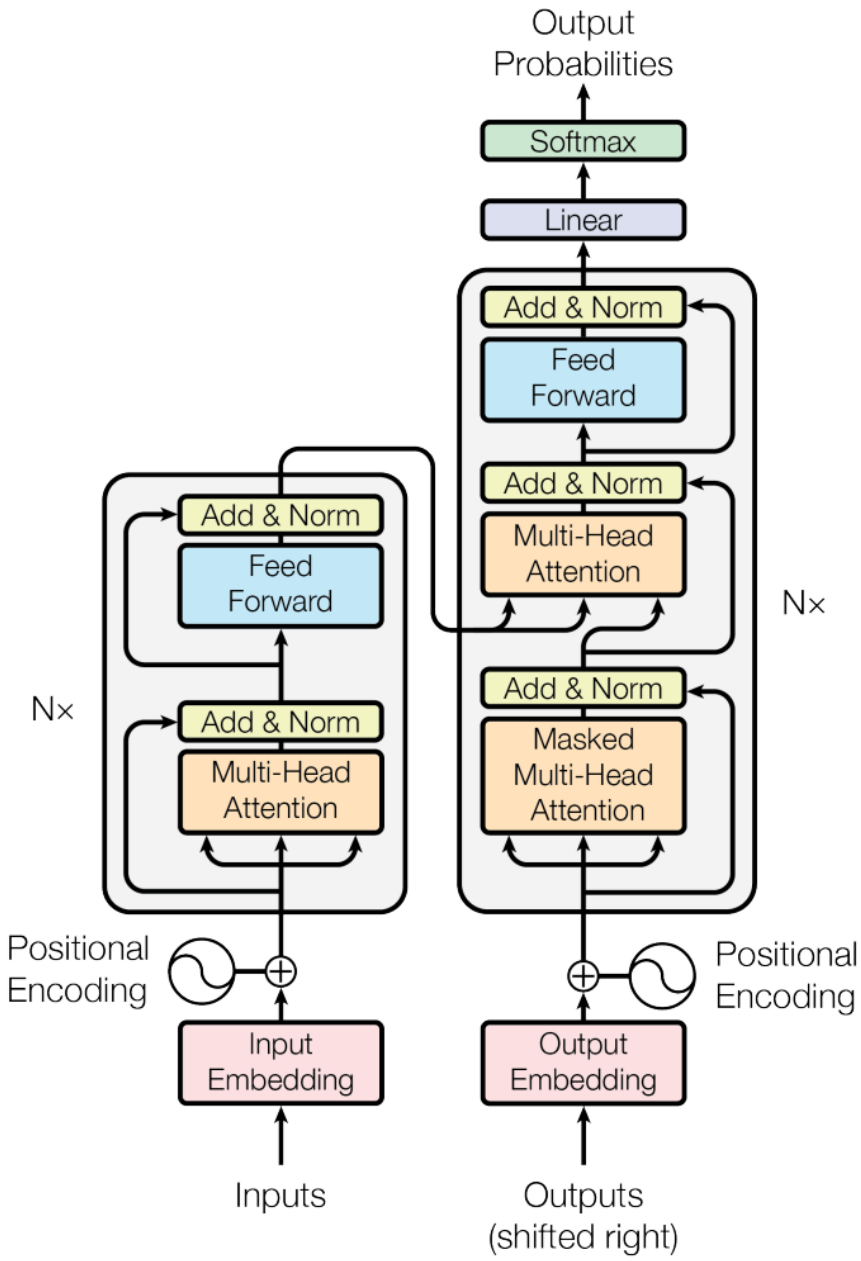


Рисунок 2.7 – Архитектура трансформера

```python
class TransformerEncoderLayer(nn.Module):
    def __init__(
        self, d_model, nhead, dim_feedforward=2048,
        dropout=0.1, relative_positional=True,
        relative_positional_distance=100,
    ):
        super(TransformerEncoderLayer, self).__init__()
        self.self_attn = MultiHeadAttention(
            d_model, nhead, dropout=dropout,
            relative_positional=relative_positional,
            relative_positional_distance=relative_positional_distance,
        )
        self.linear1 = nn.Linear(d_model, dim_feedforward)
        self.dropout = nn.Dropout(dropout)
        self.linear2 = nn.Linear(dim_feedforward, d_model)
        self.norm1 = nn.LayerNorm(d_model)
        self.norm2 = nn.LayerNorm(d_model)
        self.dropout1 = nn.Dropout(dropout)
        self.dropout2 = nn.Dropout(dropout)
        self.activation = nn.ReLU()

    def forward(
        self, src: torch.Tensor, is_causal: bool = False,
        src_mask: Optional[torch.Tensor] = None,
        src_key_padding_mask: Optional[torch.Tensor] = None,
    ) -> torch.Tensor:
        src2 = self.self_attn(src)
        src = src + self.dropout1(src2)
        src = self.norm1(src)
        src2 = self.linear2(self.dropout(self.activation(self.linear1
        ↪ (src))))
        src = src + self.dropout2(src2)
        src = self.norm2(src)
        return src
```

Рисунок 2.8 – Класс, реализующий архитектуру трансформера

Как уже говорилось ранее важной особенностью transformer является то, что он не использует слова последовательно, а использует всю последовательность сразу. Это оказывает значительное влияние на время обучения, поскольку допускает параллельную обработку, что приводит к существенному увеличению скорости, однако проблема заключается в том, что порядок слов плохо воспринимается (уровнем внимания) модели. Для решения этой проблемы необходимо ввести информацию о порядке слов во входные данные, которые будут переданы Transformer (на картинке данная процедура обозначается как «positional encoding»).

2.5.1 Механизм внимания

Вообще говоря, внимание описывает способность модели обращать внимание на важные части предложения (или изображения, или любого другого последовательного ввода). Оно делает это путем присвоения весов входным признакам на основе их важности и положения в последовательности.

Оно также играет ключевую роль в расширении контекста, который языковая модель способна учитывать при обработке и генерации языка. Это позволяет модели создавать контекстуально соответствующие и связные тексты в гораздо более длинных последовательностях.

Внимание во многом похоже на задачу поиска, когда при задании запроса q, мы хотим найти набор ключей k, наиболее похожих на q, и вернуть соответствующие значения v.

Чтобы получить вектор-строки q, k и v, представление каждого элемента входной последовательности умножается на обучаемые матрицы W_q , W_k и W_v.

Близость запроса к ключу определяется с помощью скалярного произведения:

$$\text{weights}_i = \text{softmax}\left(\frac{q_i k_1^T}{C}, \frac{q_i k_2^T}{C}, \dots\right) \tag{2.2}$$

где C – некоторая нормировочная константа. Все вышеописанные действия отображены на рисунке 2.9.

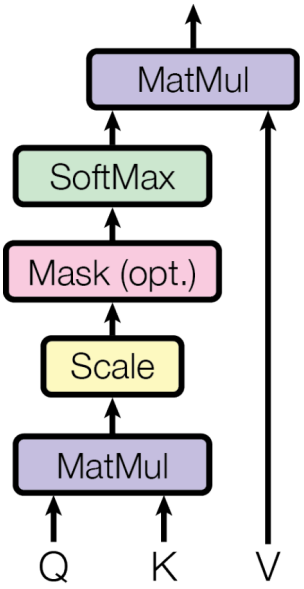


Рисунок 2.9 – Порядок операций self attention

Один набор Q, W и K может передавать лишь определённые типы взаимосвязей между токенами, ограничивая информацию, получаемую из входных представлений. Для преодоления этого ограничения, принято использовать механизм ”multi-head attention в котором несколько параллельных ”голов”внимания с различными весами применяются для агрегации разнообразных аспектов информации. Принцип работы данного механизма изображен на рисунке 2.10, а реализация показана на рисунке 2.11.

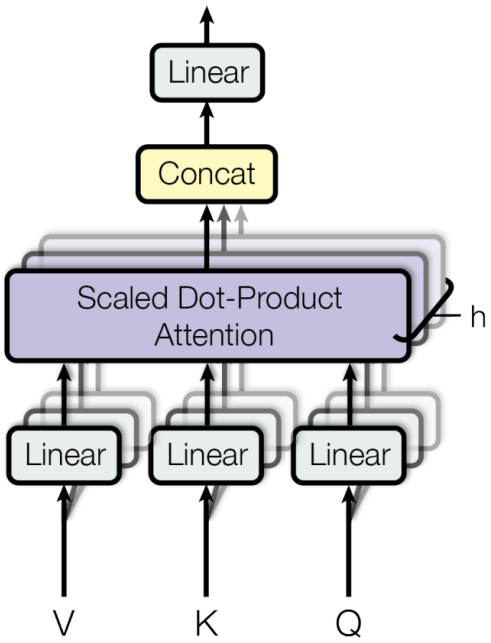


Рисунок 2.10 – Принцип работы multi-head attention

```python
class MultiHeadAttention(nn.Module):
    def __init__(self, d_model=256, n_head=4, dropout=0.1,
    ↪  relative_positional_distance=100):
        super().__init__()
        self.d_model = d_model
        self.n_head = n_head
        self.d_qkv = d_model // n_head
        self.w_q = nn.Parameter(torch.Tensor(n_head, d_model, d_qkv))
        self.w_k = nn.Parameter(torch.Tensor(n_head, d_model, d_qkv))
        self.w_v = nn.Parameter(torch.Tensor(n_head, d_model, d_qkv))
        self.w_o = nn.Parameter(torch.Tensor(n_head, d_qkv, d_model))
        nn.init.xavier_normal_(self.w_q)
        nn.init.xavier_normal_(self.w_k)
        nn.init.xavier_normal_(self.w_v)
        nn.init.xavier_normal_(self.w_o)
        self.dropout = nn.Dropout(dropout)
        self.relative_positional = LearnedRelativePositionalEmbedding
        ↪  (relative_positional_distance, n_head, d_qkv,
        ↪  True)

    def forward(self, x):
        q = torch.einsum("tbf,hfa->bhta", x, self.w_q)
        k = torch.einsum("tbf,hfa->bhta", x, self.w_k)
        v = torch.einsum("tbf,hfa->bhta", x, self.w_v)
        logits = torch.einsum("bhqa,bhka->bhqk", q, k) /
        ↪  (self.d_qkv**0.5)
        q_pos = q.permute(2, 0, 1, 3)
        l, b, h, d = q_pos.size()
        position_logits, _ = self.relative_positional(q_pos.reshape(l,
        ↪  b * h, d))
        logits = logits + position_logits.view(b, h, l, l)
        probs = self.dropout(F.softmax(logits, dim=-1))
        o = torch.einsum("bhqk,bhka->bhqa", probs, v)
        out = torch.einsum("bhta,haf->tbf", o, self.w_o)
        return out
```

Рисунок 2.11 – Класс, реализующий Multi-Head attention

2.5.2 Языковая модель

В качестве алгоритма вывода CTC будет использоваться метод поиска луча. Для его реализации была выбрана 5-граммная языковая модель, исходя из их основного достоинства, которое заключается в простоте имплементации и высокой скорости работы алгоритма.

5-грамма – это статистическая модель, которая позволяет выбирать наиболее подходящее слово по контексту на основе вероятности сочетания последовательности слов.

Для расчёта вероятности предложения используется следующее правило:

$$P(A, B) = P(B|A)P(A). \tag{2.3}$$

Для 5 слов в предложении формула будет выглядеть следующим образом:

$$P(A_1, A_2, A_3, A_4, A_5) = P(A_5|A_4, A_3, A_2, A_1) \cdot \ldots \cdot P(A_2|A_1) \cdot P(A_1). \tag{2.4}$$

Таким образом, возможно оценить общую вероятность всей цепочки, умножив условные вероятности. Однако довольно сложно точно определить вероятность слова при условии длинной последовательности предшествующих слов из-за большого числа возможных последовательностей и возможного отсутствия этих выражений в наших данных. Поэтому наилучшим вариантом является приблизительное вычисление вероятностей. А для этого используются цепи Маркова, которые позволяют предсказывать вероятность элемента последовательности, не учитывая слишком широкий контекст.

В общем виде марковская цепь k-ого порядка (когда мы учитываем контекст только последних k слов) будет выглядеть так:

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i|w_{i-k}, w_{i-k+1}, \ldots, w_{i-1}) \tag{2.5}$$

В данном случае будет использоваться марковская цепь 5-ого порядка.

Чем длиннее последовательность, тем более детализированной становится наша модель, то есть более длинные предложения содержат больше грамматики по сравнению с короткими. Одновременно с этим, чем длиннее последовательность, тем реже нам встречаются случаи употребления, что означает, что многие наблюдения возникают лишь один раз.

2.6 Метрика

В данном случае будет использоваться марковская цепь 5-ого порядка.

Чем длиннее последовательность, тем более детализированной становится наша модель, то есть более длинные предложения содержат больше грамматики по сравнению с короткими. Одновременно с этим, чем длиннее последовательность, тем реже нам встречаются случаи употребления, что означает, что многие наблюдения возникают лишь один раз.

$$\text{WER} = \frac{\text{замены} + \text{вставки} + \text{удаления}}{\text{исходная длина}} \tag{2.6}$$

2.7 Программное обеспечение

Данная работа выполнялась на операционной системе Arch linux с оконным менеджером bspwm. В качестве текстового редактора был использован nvim. Код был написан на языке программирования python с помощью библиотек, которые описаны в таблице 2.2.

Таблица 2.2 – Используемые библиотеки

№	Название	Описание
1	pytorch	платформа глубокого обучения с открытым исходным кодом, доступная с интерфейсом Python и C++
2	logging	модуль определяет функции и классы, которые реализуют гибкую систему ведения журнала событий для приложений и библиотек
3	ctcdecode	реализация алгоритма декодирования поиска луча CTC для PyTorch
4	jiwer	простой и быстрый пакет python для оценки системы автоматического распознавания речи
5	tqdm	библиотека предназначена для внедрения индикаторов выполнения во внешние интерфейсы программ на python
6	picle	модуль, который предоставляет возможность сериализовать и десериализовать объекты python

Для упрощения переносимости кода и необходимого для обучения программного окружения на другие ЭВМ, процесс обучения был проведен с помощью технологии Conda. Данная технология предоставляет готовые пакеты, позволяющие избежать необходимости иметь дело с компиляторами или пытаться понять, как именно настроить конкретный инструмент.

Среды Conda полезны для обеспечения воспроизводимости биоинформационных проектов. Полная воспроизводимость требует возможности воссоздать систему, которая изначально использовалась для получения результатов. В значительной степени этого можно достичь, используя файл среды Conda для создания среды с определенными версиями пакетов, которые необходимы в проекте.

2.8 Процесс обучения

Для обучения модели данные разбивались на две выборки: тестовую и обучающуюся. К данным выборкам применялся алгоритм фильтрации, описанный выше. После этого программа запускалась в режиме обучения, где после каждой эпохи веса модели сохранялись в отдельный файл. Само обучение заключается в подборе весов методом обратного распространения ошибки, которая высчитывалась с помощью алгоритма CTC loss.

Обучение проводится в течение 200 эпох без использования регуляризации снижения веса, график процесса обучения показан на рисунке 2.12. Скорость обучения линейно увеличивается в течение первых 1000 шагов до $3e^{-4}$, затем снижается вдвое на эпохах 125, 150 и 175. После обучения нейронная сеть была запущена на тестовой выборке, где показала точность вычисления по WER оценки в 73%. Результаты предсказаний на тестовой выборки показаны в таблице 2.3.

В данной работе также реализованы возможности запуска модели в нескольких режимах:

- обучение с нуля;
- до обучения уже готовой модели;
- тестирование готовой модели обучения.

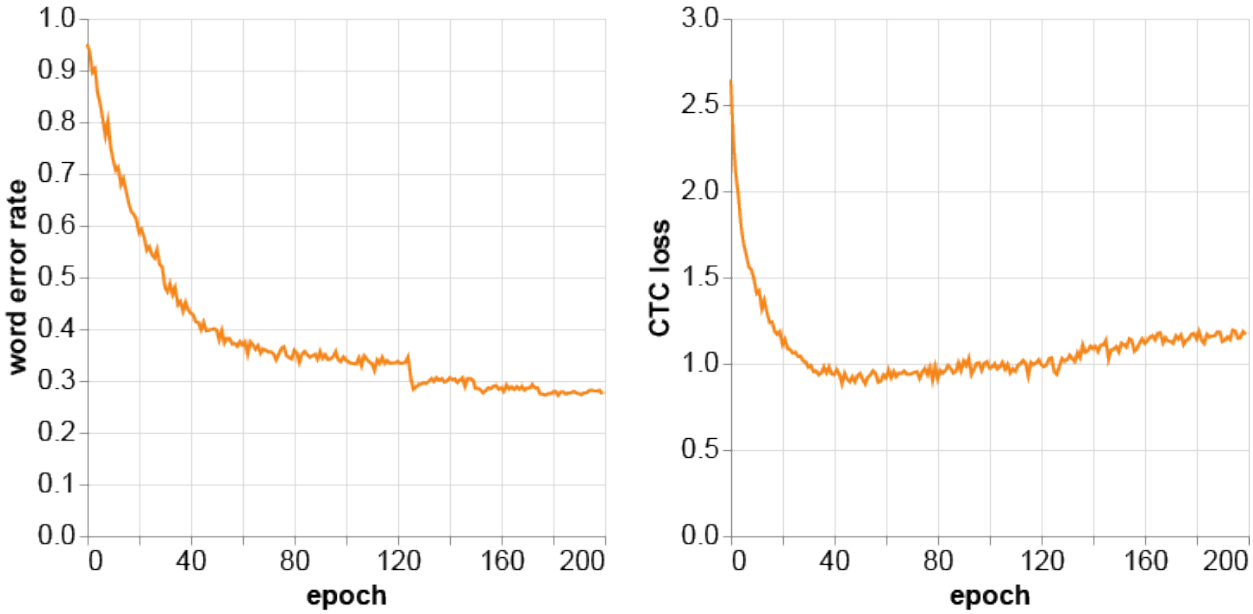


Рисунок 2.12 – График ошибок, полученных во время обучения

2.9 Пример работы программы

Таблица 2.3 – Сравнение исходных и предсказанных предложений

№	Исходное предложение	Предсказанное предложение
1	henderson he called you saw that shooting star last night	henders he called you saw that shouting star last ight
2	he read and reread the paper fearing the worst had happened to me	he read and reread the paper fearing the worst had happened to me
3	he heard footsteps running to and fro in the rooms and up and down stairs behind him	he heard footsteps running to and fro in the rooms and up out stars behind him
4	some of the refugees were exchanging news with the people on the omnibuses	some of the revenges or exciting news with the people on the unamuse
5	such things i told myself could not be	such things at old myself could not me

Продолжение таблицы 2.3

6	at the same time four of their fighting machines similarly armed with tubes crossed the river and two of them black against the western sky came into sight of myself and the curate as we hurried wearily and painfully along the road that runs northward out of halliford	at the same time for of their fighting machines simply armed with nomes across the river and over them black against the western sky came to sight of myself into the curate as we hurried whirly and painfully along the road that runs northward out of halliford
7	the hot water from the martians overthrow drifted downstream with me so that for the best part of a mile i could see little of either bank	the hot water from the martians over through driven down stream i me hussy that for the best part of a mile i could see all of crack
8	the soldiers were having the greatest difficulty in making them realise the gravity of their position	the soldiers were having the greatest difficulty in making them realise in the gravity of their position
9	their armoured bodies glittered in the sun as they swept swiftly forward upon the guns growing rapidly larger as they drew nearer	their armour bodies glittered in the sun as he swept swiftly forward upon the guns growing rapidly larger as they drew nearer
10	presently he came upon a stile and crossing it followed a footpath northeastward	presently he came upon a stile and cross it followed for both north eastward
11	all night long the martians were hammering and stirring sleepless indefatigable at work upon the machines they were making ready and ever and again a puff of greenishwhite smoke whirled up to the starlit sky	all night long the martians were hammering and stirring sleepless indefatigable at work about the machine they were making ready and after and again above of greenish white smoke whirled up to the starlit sky
12	eh said one of the men turning	eh said one of the men turning
13	suddenly he vanished and i could have fancied a faint shriek had reached me	uddenly he vanished and i could have fancied a faint shriek and reached me

Продолжение таблицы 2.3

14	there his story became ejaculatory	there storm became and jackstra
15	he is not an insurance agent	he is not and antrustion
16	it was all so real and so familiar	it was all so real and so familiar
17	i should have started at once but my companion had been in active service and he knew better than that	i should have started at once but my companion had been daniver face and he knew better the that
18	the light upon the railway puzzled me at first there were a black heap and a vivid glare and to the right of that a row of yellow oblongs	the light upon the railway must would be at first there were a black heap and a vivid glare into the right of that row of yellow blas
19	they seemed very helpless in that pit of theirs	they seemed very helpless that men of the ears
20	i myself heard nothing of that	i myself heard a thing of that
21	the authorities had reason to suppose from the size of the cylinders that at the outside there could not be more than five in each cylinderfifteen altogether	the authorities had reason to suppose from this unscaling ers that at the outside there could not be more than five in ea cylinder piety together
22	but its something more than a meteorite	but it something more than a meteorite
23	fearful massacres in the thames valley	fearful maskers and that the thames fally
24	at the end of it i sat tempering nuts with a cigarette regretting ogilvys rashness and denouncing the shortsighted timidity of the martians	at the indefessa timbering nuns with distinguee are grunting ogilvys rashness and announcing the short silent timidity of the martians
25	besides that there was quite a heap of bicycles	besides that there was quite a of my cycles
26	even then he scarcely understood what this indicated until he heard a muffled grating sound and saw the black mark jerk forward an inch or so	even then he scarcely understood with his indicated until they heard a muffled grating sound and saw the black mark jerk ford incur so

Продолжение таблицы 2.3

27	and so forth	and so forth
28	he saw this one pursue a man catch him up in one of its steely tentacles and knock his head against the trunk of a pine tree	he saw this one be a man catch him up in one of its steely in coals and tuck his head against the struck of a pine tree
29	they are dangerous because no doubt they are mad with terror	they are anserous because no doubt they a mad with error
30	i found a little crowd of perhaps twenty people surrounding the huge hole in which the cylinder lay	i found a little crowd of perhaps twenty people sound the huge hole and peto the cylinder layd

2.10 Анализ результатов

Рассмотрим результат обучения построенной модели. По WER оценки точности распознавания достигает 73%. Это показывает, что система все еще подвержена ошибкам в распознавании ЭМГ сигналов, однако полученный результат можно считать приемлемым.

Обученная нейронная сеть почти безошибочно распознает короткие и простые по грамматике предложения. Однако, чем длиннее предложение, тем хуже будет точность распознавания. Также точность распознавания ухудшается если используется необычные словосочетания. Основными причинами данных проблем являются использование статической языковой модели, а также недостаточное количество эпох и данных.

С учетом данных недостатков обученной модели, наиболее эффективным ее применением является использование для распознавание тихой речи на ограниченном словаре или для простых предложений. В остальных же случаях полученная нейронная сеть может быть подвержена ошибкам.

ЗАКЛЮЧЕНИЕ

Основной результат выпускной квалификационной работы заключается в разработке и построении с использованием современных технологий модели машинного обучения, способной переводить электромиографические сигналы речевых артикуляторов в текстовый формат.

Архитектура созданной модели основана на использовании сверточных нейронных сетей для расширения признакового пространства, а также на применении алгоритма CTC в сочетании с энкодером трансформера. Помимо этого в разработанной модели была использована 5-граммная языковая модель, что позволило повысить точность предсказания слов в предложении, а также снизить количество ошибок в словах.

В ходе выполнения данной работы были выполнены следующие задачи:

- рассмотрены подходы к обработке и анализу сигналов ЭМГ;
- проведен обзор методов распознавания речи;
- исследованы методы выделения признаков из сигналов ЭМГ;
- разработан алгоритм предварительной обработки сигналов ЭМГ;
- построена и обучена модель машинного обучения для распознавания тихой речи;
- проведен анализ полученных результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Kobrinski A., Bolkovitin S., Voskoboinikova L. Problems of bioelectric control. // Automatic and Remote Control : proc. 1st IFAC Int. Congress. — 1960. — Vol. 2. — P. 619.

2. Ганин И.П., Каплан А.Я. Интерфейс мозг компьютер на основе волны p300: предъявление комплексных стимулов “подсветка + движение”. // Журнал высшей нервной деятельности им. И. П. Павлова. — 2014. — Т. 64, № 1. — С. 32–40.

3. Унанян Н.Н., Белов А.А. Распознавание мышечной активности с помощью электромиографических датчиков в задачах управления бионическим механизмом. // Материалы 15-й Международной конференции «Устойчивость и колебания нелинейных систем управления» (конференция Пятницкого). — М.: ИПУ РАН, 2020. — С. 427–430.

4. Будко Р.Ю., Чернов Н.Н., Будко А.Ю. Распознавание мышечных усилий по сигналу лицевой электромиограммы в режиме реального времени. // Приборостроение, метролония и информационно-измерительные приборы и системы. Научный вестник НГТУ. — 2018. — Т. 71, № 2. — С. 59–74.

5. Шелухин О.И., Лукьянцев Н.Ф. Цифровая обработка и передача речи: Учебное пособие. — М.: Радио и связь, 2000. — 454 с.

6. А. С. Колоколов. Обработка сигнала в частотной области при распознавании речи. // Проблемы управления. — 2006. — С. 13–18.

7. Жиляков Е.Г., Белов С.П., Прохоренко Е.И. Вариационные методы частотного анализа звуковых сигналов. // Труды учебных заведений связи. — СПб, 2006. — № 174. — С.163-170.

8. Pascanu R., Mikolov T., Bengio Y. Understanding the exploding gradient problem. [Electronic resource]. // arXiv preprint arXiv:1211.5063. — 2012. — URL: `https://arxiv.org/pdf/1211.5063` (дата обращения: 07.02.2024).

9. Винцюк Т.К. Анализ, распознавание и интерпретация речевых сигналов. // Институт кибернетики им. В. М. Глушкова. — Киев: Наукова думка, 1987. — С. 264.

10. Hannun A., Case C., Casper J. Deep speech: Scaling up end-to-end speech recognition [Electronic resource]. // arXiv preprint arXiv:1412.5567. — 2014. — URL: https://arxiv.org/pdf/1412.5567 (дата обращения: 07.02.2024).

11. Amodei D., Anubhai R., Battenberg E., Case C. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin [Electronic resource]. // arXiv preprint arXiv:1512.02595. — 2015. — URL: https://arxiv.org/pdf/1512.02595 (дата обращения: 10.03.2024).

12. Graves A. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // Proceedings of the 23rd international conference on Machine learning. — 2006. — P. 369–376.

13. Hori T., Watanabe S., Zhang Y., Chan W. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM [Electronic resource]. // arXiv preprint arXiv:1706.02737. — 2017. — URL: https://arxiv.org/pdf/1706.02737 (дата обращения: 1.09.2023).

14. Sak H., Senior A., Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling [Electronic resource]. // Fifteenth annual conference of the international speech communication association. — 2014. — URL: https://arxiv.org/pdf/1402.1128 (дата обращения: 05.02.2024).

15. Chorowski J. Attention-based models for speech recognition. // Advances in neural information processing systems. — 2015. — P. 577-585.

16. Jozefowicz R., Vinyals O., Schuster M., Shazeer N., Wu Y. Exploring the Limits of Language Modeling [Electronic resource]. // arXiv preprint arXiv:1602.02410. — 2016. — URL: https://arxiv.org/pdf/1602.02410 (дата обращения: 05.02.2024).

17. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L. Attention Is All You Need [Electronic resource]. // arXiv preprint arXiv:1706.03762. — 2023. — URL: https://arxiv.org/pdf/1706.03762 (дата обращения: 05.02.2024).