МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное учреждение науки
«Санкт-Петербургский Федеральный исследовательский центр
Российской академии наук» (СПб ФИЦ РАН)

И.С. Кипяткова, А.А. Карпов, С.В. Кулешов,
А.А. Зайцева

МЕТОДЫ И МОДЕЛИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ

Учебное пособие

Санкт-Петербург
2021

УДК 004.522
ББК 32.813
К42

Рецензенты:
Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)», профессор *В.В. Геппенер*

Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук», директор СПИИРАН *В.Ю. Осипов*

Утверждено Ученым советом СПб ФИЦ РАН
в качестве учебного пособия

Кипяткова И.С., Карпов А.А., Кулешов С.В., Зайцева А.А.

Методы и модели автоматического распознавания речи: Учеб. пособие /СПб ФИЦ РАН. СПб., 2021. 116с.: ил. 45
ISBN 978-5-6047036-0-1

В учебном пособии рассматриваются основные методы моделирования естественной речи, применяемые в системах автоматического анализа и распознавания речи. Описываются методы и алгоритмы цифровой обработки речевых сигналов, построения фонематического словаря системы распознавания речи, акустического и языкового моделирования с использованием вероятностных и нейросетевых подходов. Дается описание интегрального подхода к построению систем автоматического распознавания речи.

Учебное пособие предназначено для студентов технических ВУЗов, аспирантов, обучающихся по направлению «09.06.01 Информатика и вычислительная техника», а также для всех читателей, интересующихся современными подходами к разработке систем автоматического распознавания речи.

УДК 004.522
ББК 32.813

ISBN 978-5-6047036-0-1

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Разработка систем автоматического распознавания слитной речи является актуальной задачей искусственного интеллекта. Речевые интерфейсы во многом более удобны для управления компьютерными и автоматизированными системами, чем стандартные графические интерфейсы. Использование речевого интерфейса позволяет пользователю одновременно выполнять несколько функций, не связанных с устройствами ввода в машину, поскольку руки пользователя остаются свободными и могут выполнять другие действия. Кроме того, системы автоматического распознавания речи могут использоваться для компьютерного стенографирования, автоматического перевода с одного языка на другой, а также в автоматизированных call-центрах и справочных системах.

В данном учебном пособии описаны основные методы и подходы, применяемые при разработке систем автоматического распознавания речи, начиная от стандартных методов (применение скрытых марковских моделей для акустического моделирования и *n*-грамм для языкового моделирования) и заканчивая современными нейросетевыми подходами.

В первой главе учебного пособия представлен основной подход к автоматическому распознаванию речи. Приведена базовая архитектура системы распознавания слитной речи. Дается классификация систем распознавания речи по размеру словаря, необходимости предварительной настройки на голос пользователя и типу распознаваемой речи. Описываются основные уровни обработки речевого сигнала. Приведены основные критерии и показатели оценки качества распознавания речи. Анализируются основные системы и инструментарии для автоматического распознавания речи.

Во второй главе представлены основные математические модели, применяемые для распознавания речи. Описываются скрытые марковские модели и основные типы искусственных нейронных сетей.

В третьей главе приводятся основные методы лексического моделирования речи. Описывается процесс создания как базового словаря системы распознавания речи, так и расширенного словаря, учитывающего вариативность произношения слов в разговорной речи. Приводится пример

комбинированного метода моделирования вариативности произношения.

Четвертая глава посвящена акустическому моделированию речи. Рассматривается процесс параметрического представления сигнала. Дается описание акустического моделирования речи с использованием скрытых марковских моделей и искусственных нейронных сетей.

В пятой главе дается описание языкового моделирования речи. Приведен обзор методов статистического моделирования разговорной речи на основе разновидностей *n*-грамм. Анализируются расширенные модели, основанные на классах слов, триггерные, морфемные, модели, дальнодействующие, факторные и нейросетевые модели.

Шестая глава посвящена методам интегрального распознавания речи. Представлены основные особенности интегральных моделей. Анализируется модель на основе коннекционной временной классификации, модель кодер-декодер, а также модель на основе архитектуры трансформер. Приведены основные подходы к улучшению качества интегральных систем распознавания речи при недостатке обучающих данных.

Учебное пособие предназначено для студентов технических ВУЗов, аспирантов, обучающихся по направлению «09.06.01 Информатика и вычислительная техника», а также для всех читателей, интересующихся современными подходами к разработке систем автоматического распознавания речи.

1 АВТОМАТИЧЕСКОЕ РАСПОЗНАВАНИЕ РЕЧИ. ОСНОВНЫЕ ПОНЯТИЯ

1.1 Базовый подход к автоматическому распознаванию речи

Под автоматическим распознаванием речи будем понимать представление непрерывного речевого сигнала, поступающего от диктора через микрофон, последовательностью слов, которая ему соответствует [74, 77]. Распознавание естественной речи – одна из основных задач распознавания образов. Задачей распознавания образов является отнесение входного образа к отдельным существующим классам. Возможные классы моделируются в зависимости от применения (типа анализируемых данных). Целью распознавания является оценка выходного класса (выходной гипотезы), к которому входные данные принадлежат с наибольшей вероятностью.

Задача системы распознавания речи заключается в том, чтобы по речевому сигналу правильно идентифицировать сказанную диктором последовательность слов. Это соответствует оптимальному критерию, который может быть выражен как [44]:

$$\hat{w} = \arg\max_{w \in \mathbf{W}} P\left(w \mid \mathbf{O}\right), \tag{1}$$

где $\hat{w}$ – выходная гипотеза фразы, w – произнесенная последовательность слов, $\mathbf{W}$ – набор всех возможных последовательностей слов (гипотез), $\mathbf{O}$ – последовательность векторов признаков, вычисленных по входному речевому сигналу.

После применения формулы Байеса формула (1) принимает следующий вид:

$$\hat{w} = \arg\max_{w \in \mathbf{W}} \frac{P\left(\mathbf{O} \mid w\right) P\left(w\right)}{P\left(\mathbf{O}\right)}$$

$P(\mathbf{O})$ не изменяется в зависимости от последовательности сказанных слов w, поэтому $P(\mathbf{O})$ можно пренебречь. Таким образом, критерий максимума апостериорной вероятности будет иметь следующий вид:

$$\hat{w} = \arg\max_{w \in \mathbf{W}} P(\mathbf{O} \mid w) P(w),$$

где $P(\mathbf{O}|w)$ – вероятность того, что текущий вектор признаков **O** наблюдается, если диктором произносится последовательность слов *w*. Это выражение называется акустической вероятностью и вычисляется с помощью моделей декодера речевого сигнала. $P(w)$ – априорная вероятность появления некоторого слова во фразе, которая вычисляется с помощью языковых моделей.

На рисунке 1.1 показана схема базовой системы автоматического распознавания речи.

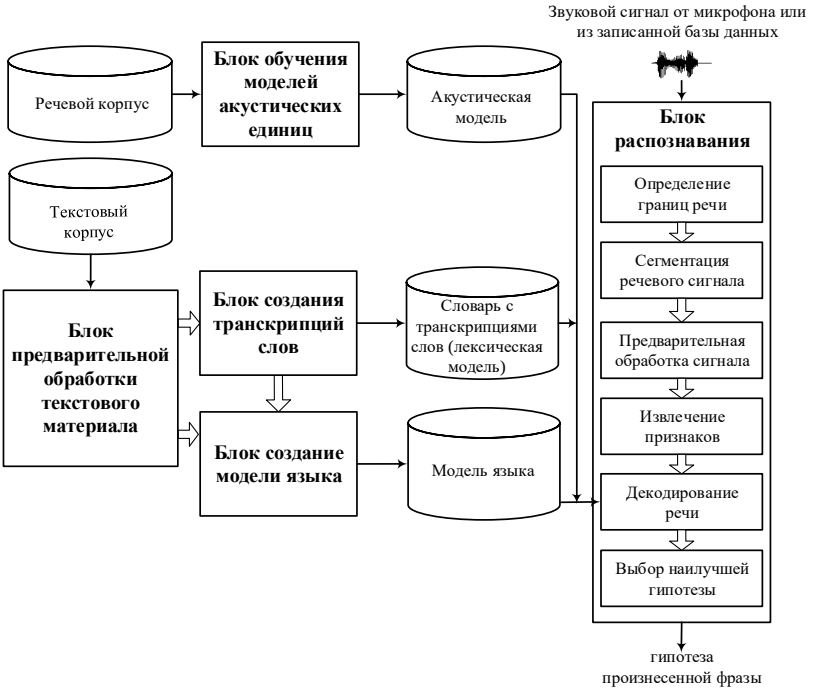


Рис. 1.1. Базовая архитектура системы распознавания слитной речи

Система работает в двух режимах: обучение и распознавание. В первую очередь выполняется обучение акустической и языковой моделей. Акустические модели обучаются по векторам

признаков, вычисленным по речевым данным и их известным транскрипциям. Модель языка строится путем оценивания частоты встречаемости комбинаций слов в тексте предметной области. При распознавании входной сигнал преобразуется в последовательность векторов признаков, и с помощью предварительно обученных моделей звуков речи (фонем) и модели языка или грамматики производится поиск наиболее вероятной гипотезы или подмножества лучших гипотез [44].

1.2 Классификация систем распознавания речи

Рассмотрим несколько вариантов классификации систем распознавания речи. Одним из основных критериев классификации является размер распознаваемого словаря (см. таблицу 1.1). Согласно принятой в мире классификации [5], малым словарем распознавания считается словарь в единицы и десятки слов, средний распознаваемый словарь содержит сотни слов, большой словарь содержит тысячи и десятки тысяч слов, словарь размером в сотни тысяч и миллионы слов считается сверхбольшим [61], словарь, который пытается моделировать все существующие и потенциально возможные слова в некотором языке считается неограниченным (unlimited) [57].

Таблица 1.1

Классификация словарей распознавания по размеру

Размер словаря распознавания	Количество слов, V
Малый	$V<100$
Средний	$100 \leq V < 1000$
Большой	$1000 \leq V < 100000$
Сверхбольшой	$V \geq 100000$
Неограниченный	$V \to \infty$

В зависимости от необходимости предварительной настройки на голос пользователя различают дикторозависимые и диктронезависимые системы распознавания речи. В некоторых приложениях с малым и средним словарем предпочтительны дикторозависимые системы (например, с целью недопущения несанкционированного доступа), основанные на сравнении входного сигнала с ранее записанными эталонами слов и

фраз [78]. При этом пользователь обычно вносит в память системы полный набор образцов слов заданного словаря, что можно считать этапом предварительного обучения системы на акустическом уровне. С ростом словаря время обучения (надиктовки эталонных слов/фраз) возрастает линейно.

Дикторонезависимые системы распознавания речи могут работать без настройки на голос пользователя, то есть любой диктор сразу может вступить в диалог с системой, не проводя процесс обучения системы, либо с частичной подстройкой, когда диктор произносит не весь словарь, а некоторый адаптационный текст (что существенно при большом объеме словаря), чтобы система могла настроить необходимые акустические базы данных. В таких системах дикторонезависимость достигается путем статистического моделирования языковых и речевых процессов, а для этого необходимы значительные объемы акустических данных, позволяющих создавать стохастические модели. После того как система обучена на речевом корпусе, она должна достаточно точно распознавать речь среднестатистического диктора, представленного этим корпусом.

Модели распознавания речи во многом зависят от используемого вида (стиля) речи: изолированная речь, слитная, прочитанная и разговорная (спонтанная речь, используемая, например, в диалогах является наиболее сложным случаем разговорной речи) [45]. Задача распознавания изолированной речи является наиболее простой и уже решена для малого словаря. Элементами словаря являются модели слов или целых фраз, а также модель паузы, которая необходима для определения границ речи. Возможные слова моделируются распознавателем как отдельные классы. Когда речевой сигнал должен быть распознан, распознаватель вычисляет оценки для каждого возможного класса, и класс с наивысшей оценкой выбирается в качестве лучшей выходной гипотезы. Качество распознавания здесь целиком зависит от размера словаря. С одной стороны, большое количество слов и наличие разных вариантов произношения компенсируют неточности со стороны говорящего, а с другой стороны, увеличение словаря ведет к возникновению близких по звучанию слов, что снижает точность распознавания. При вводе изолированной речи пользователь

должен делать искусственные паузы между словами или же говорить слитно и точно жестко заданные фразы, которые известны системе.

Распознавание слитной речи является более сложной задачей, при которой распознаются отдельные фразы, составленные из ключевых слов, хранящихся в словаре. Качество подобных систем зависит от полноты словаря и модели языка, которая задает правила связи слов в распознанной фразе. Распознаватели слитной речи рассматривают речевой сигнал как последовательность связанных друг с другом и осмысленных слов. Каждое возможное слово моделируется как отдельный класс, и речевому сигналу приписывается последовательность классов (слов) с наибольшей оценкой. Эта процедура требует моделирования каждого возможного слова в словаре.

Прочитанная речь широко используется и для обучения, и для тестирования систем распознавания речи, потому что ее гораздо проще и дешевле записать и обработать, чем спонтанную речь. Подобный стиль речи имеет постоянный темп и артикуляцию (четкое проговаривание звуков). В прочитанной речи, как правило, нет запинок, таких как куски слов, заикание, самоисправления, так как подобные элементы, если и случаются, то удаляются из речевого корпуса на этапе обработки. В основном используются фонетически представительные короткие и простые высказывания. Нет озвученных пауз, фальш-стартов, исправлений или слишком длинных пауз. Соответственно, не наблюдается также эмоциональных и смысловых ударений внутри фразы. Таким образом, для прочитанной речи характерны монотонность, отсутствие интонации и четкость произношения фонем (звуков речи).

Разговорная речь обладает множеством особенностей, которые не учитываются в других типах речи, и которые способны существенно ухудшить качество распознавания для системы, обученной не на спонтанной речи. Темп речи не постоянен и внутри высказывания, и между высказываниями в ходе сессии записи, и между сессиями, и для разных дикторов. Артикуляция также сильно меняется, ударные формы для важных слов и безударные формы для контекстных слов не наблюдаются в читаемой речи. Длина предложений увеличивается. В спонтанной речи сильно варьируется

произношение слов, возникают явления редукции и ассимиляции звуков, для учета которых необходимо создавать альтернативные транскрипции слов и, тем самым, расширять словарь произношений.

1.3 Уровни обработки речевого сигнала

Общая схема обработки естественного языка, на которой выделено несколько уровней описания речевой коммуникации, представлена на рисунке 1.2 [42].

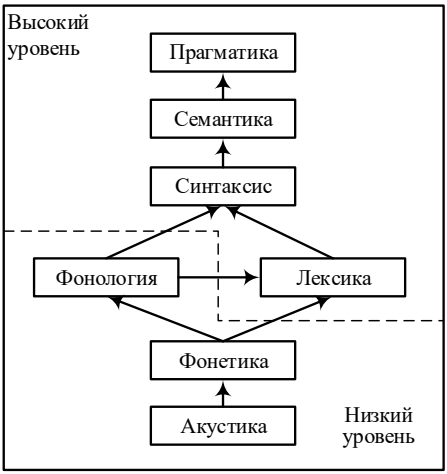


Рис. 1.2. Основные уровни обработки речи

В первую очередь следует заметить, что выделяются два базовых уровня представления и обработки речи: высокий и низкий. Чем ниже уровень, тем ближе к физическому описанию звукового сигнала. Подобная классификация от высокуровневого к низкоуровнему описанию применима ко всем типам взаимодействия и процессам, поскольку всегда можно выделить физические сигналы и их смысловую интерпретацию. Рассмотрим далее, какие уровни обработки и какие виды знаний следует учитывать при моделировании речевой коммуникации [89].

Акустический уровень является самым низким, поскольку касается исключительно аудиосигнала. Речь — это, в первую очередь, последовательность звуков, которые создаются за счет

изменения давления воздуха посредством голосового тракта. Изучение акустических процессов включает в себя запись, цифровую обработку, параметризацию акустического сигнала [19, 53]. В общем случае, акустический уровень занимается методами кодирования, параметрического представления речевого сигнала.

На *фонетическом уровне* в центре внимания находятся свойства речевого сигнала на уровне звуков, порожденных артикуляционной системой [72]. Фонетика изучает, как люди управляют мышцами голосового тракта для изменения его геометрических характеристик с помощью языка, губ, зубов и других органов при произнесении конкретных звуков речи.

Фонологический уровень — это первый уровень, на котором появляются некоторые семантически значащие единицы. В то время как фонетика имеет дело с естественным порождением различных звуков, фонология занимается анализом ограниченного числа отдельных звуков возможных в определенном языке (фонемы), ритма, с которым они порождаются в высказывании, мелодичности, используемой в конкретной фразе, частоты основного тона каждой произнесенной фонемы и смыслового ударения во фразе. В некоторых языках, например в китайском, изменения интонации может полностью изменить значение слова. Этот уровень является переходным между высокими и низкими уровнями, поскольку здесь одновременно приходится учитывать параметрическое представление речевого сигнала, специфические особенности языка и просодические характеристики.

Лексический уровень. Как было сказано ранее, в каждом языке существует конечное число различных звуков (фонем). Тем не менее невозможно произнести произвольную последовательность звуков определенного языка и породить тем самым значащее слово. Поэтому на лексическом уровне производится описание всех значащих последовательностей фонем, которые обозначают слова. На данном уровне единицами описания являются слова или части слов (морфемы), высокоуровневая обработка ведется уже без привлечения физического аудиосигнала – только с текстом. На рисунке 1.2 показано, что фонология связана как с лексикой, так и с

фонетикой, поскольку в определенных языках (таких, как китайский), различные просодические модели применяют к сходным последовательностям фонем, порождая различные слова, а следовательно, и различный смысл.

Синтаксический уровень. По аналогии с тем, что не всякая последовательность фонем может являться словом, следует отметить, что и не всякая последовательность слов может являться предложением, и существует некоторый набор правил, который ограничивает допустимые цепочки слов. Набор грамматических или стохастических правил, описывающих возможные комбинации слов, составляет суть синтаксического уровня. Для каждого языка существует свой набор грамматик, который определяет функцию каждого слова в предложении и, таким образом, описывает его синтаксическую структуру. В общем случае, грамматические правила, описанные строгой лингвистикой, как ее учат в школе, характерны для письменной речи, но не подходят для моделирования устной речи. Поэтому при автоматическом анализе устной речи грамматические правила следует использовать осторожно, поскольку они накладывают жесткие ограничения, в результате чего понятные (но содержащие некоторые неточности) фразы могут быть отвергнуты системой. Уже в первых работах по вычислительной лингвистике были подняты проблемы несоответствия устной и письменной речи, и необходимости поиска более гибких «мягких» подходов к оценке связей и структуры внутри предложения [10, 65, 86].

Семантический уровень. Даже если высказывание синтаксически правильно, нет гарантии, что оно несет в себе осмысленную информацию. Поэтому следующим шагом описания речевой коммуникации является проверка осмысленности предложения и извлечение смысла. На семантическом уровне начинается исследование контекстно-независимого значения слов, а также их комбинаций в предложении.

Прагматический уровень. Прагматика собирает и оценивает в целом всю контекстно-зависимую информацию в процессе коммуникации. Здесь участвует не только текущая информация, получаемая в процессе текущего речевого акта, но и все знания, которые имеют участники диалога, и даже окружающие условия.

Бывает, что собеседники не сразу понимают друг друга, если говорят о разных вещах, хотя при этом воспринимают все слова и даже правильную грамматическую структуру фразы. Такие ситуации часто возникают при быстром переходе с темы на тему в ходе разговора или несоответствии общих знаний у собеседников. Эти знания, прежде всего, связаны с контекстом или социо-культурным положением индивида.

Необходимо отметить, что большинство современных систем автоматического распознавания речи имеют многоуровневую иерархическую структуру, учитывающую разного рода сведения о языке и текущей ситуации. При недостатке информации или зашумленных окружающих условиях на выходе каждого уровня формируется не единственное решение (например, одна цепочка фонем), а несколько возможных гипотез, выбор между которыми производится на следующем уровне обработки.

1.4 Критерии и показатели оценки работы системы автоматического распознавания речи

Одним из важных вопросов автоматического распознавания речи является объективное количественное оценивание результатов распознавания, которое имеет важное значение как для разработчиков, так и для конечных пользователей систем. Методология оценки производительности необходима для сравнения и сопоставления различных систем распознавания и в ней различают критерий, показатель и метод. Критерий — это правило принятия решения по оценке чего-либо на соответствие предъявленным требованиям. Показатель (мера или метрика) определяет конкретное свойство, которое оценивается для выбранного критерия оценки (например, процент правильно распознанных слов, время обработки сигнала, уровень максимально допустимого шума при сохранении работоспособности и т. п.) [75]. Часто показатель ассоциируют с целевой функцией. Метод – это способ определения соответствующего значения для данного показателя (сравнение распознанных слов с последовательностью сказанных слов, оценка времени обработки в секундах и т. п.).

Системы распознавания речи оценивают по качеству распознавания и по скорости распознавания. Тестирование выполняется по речевым записям, которые не использовались

для обучения системы. Качество работы систем распознавания речи оценивается путем сравнения последовательности распознанных слов с тем, что действительно было произнесено, при этом выделяют три типа ошибок: замена одного слова другим, удаление слова, вставка слова. Таким образом, относительное количество (коэффициент) неверно распознанных слов (англ. word error rate; WER) определяется следующим образом [75]:

$$E = \frac{I + D + S}{N} \cdot 100\%,$$

где D, I, и S — количество неверно удаленных, вставленных и замененных слов соответственно, N — общее число слов в распознаваемом сообщении.

Существуют два показателя, по которым определяется качество распознавания: правильность и точность распознавания. Правильность определяется следующим образом [63]:

$$C = \frac{H}{N} \cdot 100\%,$$

где H — количество правильно распознанных слов:

$$H = N - D - S$$

Этот показатель не учитывает ошибки, связанные со вставкой слов, поэтому обычно используется другой показатель — точность распознавания, который вычисляется по следующей формуле [63]:

$$A = \frac{H - I}{N} \cdot 100\%$$

Пример

Представим, что была произнесена фраза: «*По предварительным прогнозам в северо-западном регионе России действительно станет теплее*».

В результате же автоматического распознавания была получена следующая фраза: «*По предварительным прогнозам в северо-западный регионе в российский состоится при*».

Количество слов в исходной фразе N=10. Определим количество ошибок, допущенных при распознавании. Были неправильно распознаны слова: «северо-западном», «России», «станет», «теплее». То есть количество замен S=4. Кроме того, было ошибочно вставлено слово «в» (I=1) и удалено слово «действительно» (D=1). Тогда относительное количество неправильно распознанных слов будет равно:

$$E = \frac{1+1+4}{10} \cdot 100\% = 60\%$$

Количество правильно распознанных слов:

$$H = 10 - 1 - 4 = 5$$

Правильность распознавания:

$$C = \frac{5}{10} \cdot 100\% = 50\%$$

Точность распознавания:

$$A = \frac{5-1}{10} \cdot 100\% = 40\%$$

Кроме того, иногда используется специальный показатель ошибки распознавания для флективных языков, к которым относится и русский [51]. Этот показатель ошибки распознавания приписывает вес $k_{\text{inf\_1}}$ всем ошибкам, которые привели к изменению лексемы, т.е. целого слова (грубая ошибка распознавания S_1 – замена лексемы), и вес $k_{\text{inf\_2}}$ всем ошибкам в словах, где было неверно распознано окончание словоформы, но лексема слова осталась правильной (негрубая ошибка S_2 – замена окончания). В этом случае флективная ошибка распознавания речи будет вычисляться следующим образом:

$$E_{\text{inf}} = \frac{I + D + k_{\text{inf\_1}} \cdot S_1 + k_{\text{inf\_2}} \cdot S_2}{N} \cdot 100\%$$

Весовые коэффициенты $k_{\text{inf\_1}}$, $k_{\text{inf\_2}}$, могут принимать значения от 0 до 1, при этом должно выполняться условие $k_{\text{inf\_1}} > k_{\text{inf\_2}}$.

Пример

Для приведенной выше фразы определим флективную ошибку распознавания при $k_{\text{inf\_1}}=1$, $k_{\text{inf\_1}}=0{,}5$. В трех словах ошибки распознавания привели к изменению лексемы ($S_1=3$), а одна ошибка была допущена только в окончании слова (было распознано "северо-западный" вместо "северо-западном"), то есть $S_2=1$. Тогда флективная ошибка распознавания слов во фразе будет равна:

$$E_{\text{inf}} = \frac{1+1+1\cdot 3+0{,}5\cdot 1}{10} \cdot 100\% = 55\%$$

Скорость обработки речи может оцениваться по нескольким показателям [75]. Обычно она вычисляется с использованием меры, называемой показателем скорости (англ. Speed Factor; SF), также известной как показатель реального времени (англ. Real Time; RT), который определяется как отношение общего времени обработки, требуемого для анализа всей записанной речи на одном ядре процессора, к длительности исходного анализируемого аудиосигнала. Например, если 10-минутный аудиофайл обрабатывается системой распознавания речи ровно 5 минут, то $SF=0{,}5$ реального времени, если он обрабатывается в течение 20 минут, то тогда $SF=2{,}0$ реального времени, что уже значительно хуже. Скорость обработки может быть также указана в абсолютных значениях времени (например, количество минут/секунд для обработки входного сигнала), однако это не является наглядным. Другим показателем скорости автоматического распознавания речи является период ожидания обработки отсчета (англ. Sample Processing Latency; SPL). Этот показатель означает максимальное количество аудиоданных,

которое алгоритм распознавания должен обработать до выдачи результата для первого отсчета сигнала.

При разработке системы автоматического распознавания речи со (сверх)большим словарем, работающей в реальном масштабе времени с использованием микрофона (онлайн режим), часто требуется найти компромисс между точностью распознавания и скоростью обработки. Настройка некоторых параметров для распознавания речи может улучшить точность распознавания, но уменьшить скорость обработки. В этом случае может быть полезным график, показывающий значения WER в зависимости от значений скорости распознавания (SF) для некоторых контрольных точек [24].

1.6 Современные системы и инструментарии для автоматического распознавания речи

Рассмотрим основные инструментарии для автоматического распознавания речи. Крупнейшим комплексом программных средств для разработки систем автоматического распознавания речи является Kaldi [43], написанный на языке программирования C++ на основе библиотеки OpenFST и с использованием библиотек BLAS и LAPACK, выполняющих основные операции линейной алгебры. По сравнению с другими свободно-доступными программными средствами в Kaldi реализовано наибольшее количество современных подходов к распознаванию речи, в том числе различные типы нейронных сетей. Также достоинством Kaldi является наличие большого числа готовых скриптов, реализующих различные подходы к обучению системы распознавания речи.

Комплекс программ HTK (Hidden Markov Model Toolkit) [54], разработанный в Кембриджском университете, был первым широко известным инструментарием для обучения акустических моделей и выполнения декодирования речи. Как можно увидеть из названия, HTK предназначен для работы со скрытыми марковскими моделями (CMM), однако версия 3.5 также поддерживает возможность использования нейронных сетей [67].

Еще одним крупнейшем программным комплексом для разработки систем распознавания речи является CMU Sphinx [12], разработанный в университете Карнеги-Меллона с участием Массачусетского технологического института и компании Sun

Microsystems. CMU Sphinx существенно уступает Kaldi по числу реализованных методов, не использует нейросетевые подходы. В частности, для акустического моделирования используются только СММ, а для моделей языка – *n*-граммы. Плюсом данной системы является наличие готовых акустических и языковых моделей для множества языков, включая русский.

Кроме того, некоторые компании предоставляют свои API (интерфейсы программирования приложений) для преобразования речи в текст. Одним из самых известных является Cloud Speech-to-Text API от компании Google, который поддерживает 120 языков. Модели системы построены на основе глубоких нейронных сетей. Распознавание речи выполняется удаленно на серверах Google. Данный сервис является платным, однако есть и бесплатная версия с ограниченным число запросов.

Yandex SpeechKit – сервис от компании Яндекс, лежащий в основе голосового помощника Алиса. Поддерживает три языка: русский, английский и турецкий. Построен на основе глубоких нейронных сетей. Есть возможность дообучения моделей для необходимой предметной области. Сервис является платным.

Кроме того, стоит упомянуть Microsoft Speech API, разработанный компанией Microsoft для распознавания и синтеза речи в Windows, который поддерживает около 100 языков, а также Apple Speech Kit – технологию распознавания речи, лежащую в основе голосового помощника Siri, поддерживающую более 30 языков.

1.7 Вопросы по разделу 1

1. В чем состоит задача автоматического распознавания речи?
2. Укажите возможные способы классификации систем распознавания речи.
3. Как классифицируются системы распознавания речи по размеру словаря?
4. Какие существуют типы речи?
5. Назовите основные особенности разговорной речи.
6. Назовите самый низкий уровень обработки речевого сигнала.
7. Назовите основные критерии и показатели оценки систем распознавания речи.

8. Как определяется количество неверно распознанных слов?

9. Какие ошибки распознавания относятся к негрубым при расчете флективной ошибки распознавания?

10. Назовите основные программные инструментарии для разработки систем распознавания речи.

2 ОСНОВНЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ПРИМЕНЯЕМЫЕ ДЛЯ РАСПОЗНАВАНИЯ РЕЧИ

2.1 Скрытые марковские модели

В настоящее время наиболее перспективные системы распознавания слитной речи строятся на основе скрытых марковских моделей/цепей (СММ). В основу метода положено представление о марковской цепи и марковском процессе в теории вероятностей, предложенное выдающимся петербургским математиком А.А. Марковым еще в начале XX века [83]. Рассматривается некий объект, который имеет несколько состояний и может переходить из одного состояния в другое, причем текущее состояние зависит только от предыдущего. Поведение объекта определяется цепью переходов из начального состояния в текущее. Это и есть марковская цепь. Она изображается в виде графа, где вершины обозначают состояния объекта, а направленные дуги показывают возможные пути перехода из состояния в состояние (рисунок 2.1). Возле каждой дуги проставляется вероятность этого перехода. Вероятность перехода из некоего состояния X в состояние Z через промежуточное состояние Y определяется как:

$$P(X,Y,Z) = P(X) \cdot P(Y|X) \cdot P(Z|Y)$$

где $P(Y|X)$ и $P(Z|Y)$ – условные вероятности перехода из состояния X в состояние Y и из состояния Y в состояние Z.

Скрытая марковская модель является развитием математического аппарата марковских цепей и описывает пару стохастических процессов, первый из которых не поддается прямому наблюдению и представляет собой обычную марковскую цепь, а второй наблюдаем и представляется последовательностью случайных переменных в пространстве акустических параметров [44]. В скрытой марковской модели текущее состояние точно не известно. В каждой вершине графа объект может принимать с некоторой вероятностью любое из возможных состояний (рисунок 2.2).

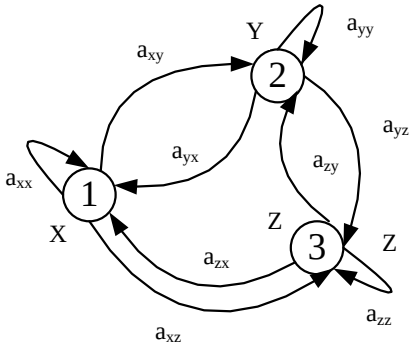


Рис. 2.1. Пример марковской цепи первого порядка

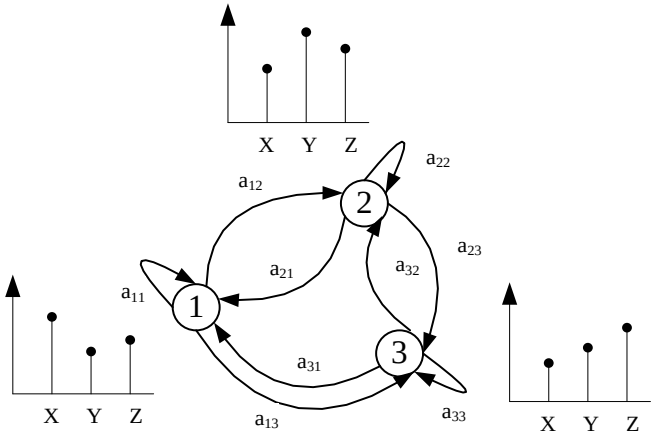


Рис. 2.2. Скрытая марковская цепь

Если для обычной марковской цепи по известной последовательности состояний можно проследить все переходы объекта (например, последовательности состояний *YXXZYX* на рисунке 2.1 однозначно соответствует цепь переходов 2-1-1-3-2-1), то для скрытой марковской цепи это невозможно, то есть цепь переходов скрыта за стохастическим процессом [81]. В этом случае имеет место двойная стохастичность: не только переходы из состояния в состояние случайны, но даже и принимаемое в результате перехода состояние определяется распределением вероятностей. В случае распознавания речи двойная

стохастичность нужна для представления двух видов изменчивости речевого сигнала – временной и спектральной. Временная изменчивость моделируется вероятностными переходами, а спектральная изменчивость – вероятностными состояниями скрытой марковской цепи.

2.2 Искусственные нейронные сети

2.2.1 Основные понятия

Искусственные нейронные сети (ИНС) представляют собой набор математических и алгоритмических методов для решения широкого круга задач [90]. Структурным элементом сети является искусственный нейрон, построенный по принципу биологического прототипа. Искусственные нейронные сети — это совокупность моделей биологических нейронных сетей, состоящих из искусственных нейронов, связанных между собой синаптическими соединениями. Работа сети состоит в преобразовании входных сигналов во времени, в результате чего меняется внутреннее состояние сети и формируются выходные воздействия [70].

Параметры модели ИНС определяются в процессе машинного обучения. Машинное обучение (англ. machine learning) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач [37]. Основная идея машинного обучения состоит в том, чтобы научить машину обучаться. Для этого необходимо дать машине некоторые примеры, т.е. некоторые выборки данных. На основе этих данных машина должна обучить модель таким образом, чтобы выполнять предсказания на основе этих данных.

Таким образом, основная идея обучения нейронной сети состоит в том, что есть некоторый обучающий набор данных, состоящий из набора примеров, которые представлены вектором $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, при этом $\boldsymbol{X}_1$ – вектор признаков для первого примера, $\boldsymbol{X}_2$ – вектор признаков для второго примера и т.д., $\mathbf{X}_N$ – вектор признаков для примера N. Для каждого примера есть связанная с ним метка: $y_1, y_2, \ldots, y_N$. Необходимо построит модель, которая по входным данным предсказывала бы соответствующую им метку, называемую также целевой

переменной. При этом важно, чтобы модель позволяла получить подобное предсказание для будущих примеров данных, т.е. не для тех данных, которые входили в обучающий набор, а для других данных, для которых истинная метка неизвестна.

2.2.1 Модель искусственного нейрона

Для начала рассмотрим простейшую модель одиночного нейрона, процесс обучения которого показан на рисунке 2.3.

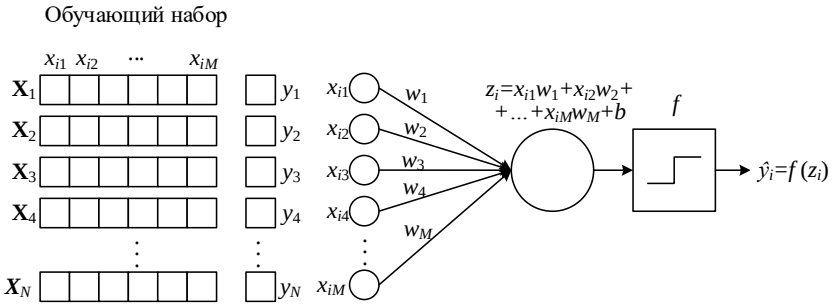


Рис. 2.3. Процесс обучения модели искусственного нейрона

В примере, представленном на рисунке 2.3 используется N обучающих примеров, $\mathbf{X}$ – это данные, $\mathbf{X}_i$ – соответствует i-му примеру, x_{i1} – соответствует первому компоненту вектора $\boldsymbol{X}_i$, x_{i2} – соответствует второму компоненту вектора $\mathbf{X}_i$, и т.д., x_{iM} – соответствует компоненту с индексом M вектора $\mathbf{X}_i$, $\mathbf{W} = \{w_1, w_2, \ldots, w_M\}$, b – параметры модели. Выход модели определяется следующим образом. Каждый компонент вектора $\mathbf{X}$ умножается на соответствующий параметр модели $\mathbf{W}$, называемый весом. Таким образом, параметр w_1 будет умножаться на первый компонент вектора $\mathbf{X}_i$ (x_{i1}), параметр w_2 будет умножаться на второй компонент вектора $\mathbf{X}_i$ (x_{i2}), параметр w_M будет умножаться на компонент вектора $\mathbf{X}_i$ с индексом M (x_{iM}). Результаты всех произведений поступают на суммирующий блок. Таким образом, для каждого i-го обучающего примера вычисляется взвешенная сумма (z_i):

$$z_i = (w_1 \cdot x_{i1}) + (w_2 \cdot x_{i2}) + \ldots + (w_M \cdot x_{iM}) + b,$$

где b – константа, называемая смещением (англ. bias), которая нужна для того, чтобы иметь возможность изменять порог возбуждения нейрона.

Затем результат суммирования (z_i) преобразуется активационной функцией ($f(z_i)$) и дает выходной сигнал ($\hat{y}_i$). Активационная функция может быть пороговой и принимать значения 0 или 1 в зависимости от пороговой величины, линейной функцией, а также нелинейной, что дает большие возможности нейронной сети (основные виды активационных функций будут рассмотрены в следующем разделе).

Цель обучения модели – подобрать параметры модели (веса и смещение) так, чтобы они согласовывались с обучающими данными.

2.2.2 Основные виды активационных функций

Самый простой вид активационной функции - единичный скачок или жесткая пороговая функция (см рисунок 2.4а). При z больше или равном нулю выход равен 1, если z меньше нуля, то выход равен нулю:

$$y = f(z) = \begin{cases} 1, & \text{при } z \geq 0 \\ 0, & \text{при } z < 0 \end{cases}$$

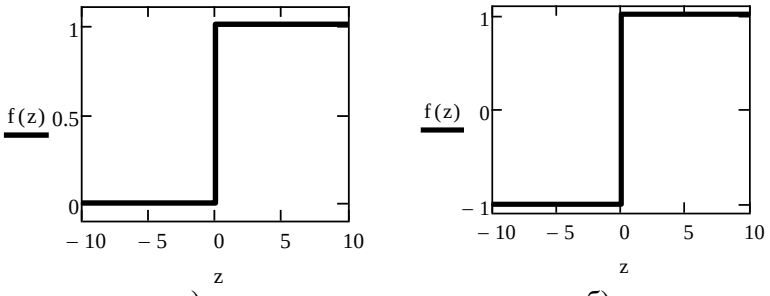


Рис. 2.4. Активационные функции: а) единичный скачок, б) знаковая функция

У данной функции есть разновидность – знаковая функция, данная функция симметрична относительно нуля, график

функции показан на рисунке 2.4б. Выход данной функции также равен 1 при z больше или равном нулю и равен -1 при отрицательном z:

$$f(z) = \begin{cases} 1, & \text{при } z \geq 0 \\ -1, & \text{при } z < 0 \end{cases}$$

Одной из наиболее широко используемых активационных функция является сигмоидальная (или логистическая) функция, показанная на рисунке 2.5а. Значение сигмоидальной функции всегда находится в интервале от 0 до 1.

$$f(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$

На рисунке 2.5б показан гиперболический тангенс, он похож по форме на сигмоидальную функцию, но симметричен относительно нуля:

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Если на выходе сигмоидальной функции значения от 0 до 1, то на выходе гиперболического тангенса – от -1 до 1.

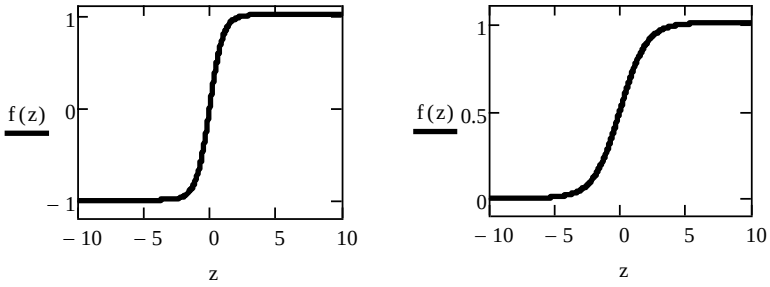


Рис. 2.5 – Активационные функции: а) сигмоидальная, б) гиперболический тангенс

Как сигмоидальная функция, так и гиперболический тангенс очень широко используются, однако у них есть недостаток, как можно видеть на графиках, у них очень большие практически плоские области, т.е. в этих областях большое изменение значения z приведет к очень маленькому изменению выхода функции. Этого недостатка лишена другая широко применяемая активационная функция – ReLU (Rectified Linear Unit), график которой приведен на рисунке 2.6.

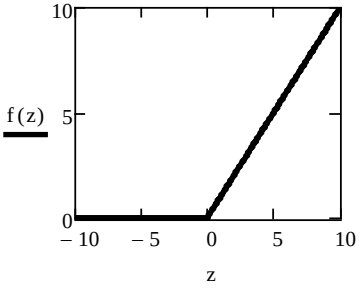


Рис. 2.6. Активационная функция ReLU

Значение функции ReLU равно нулю для любого отрицательного значения z (т.е. если общая сумма входных данных отрицательна, то значение функции равно нулю), если значение входных данных больше нуля, то она возвращает это значение, т.е. z. Иными слова значение функции определяется как максимальное значение между суммой входных данных и нулем:

$$f(z) = \max(0, z) = \begin{cases} z, & \text{при } z \geq 0 \\ 0, & \text{при } z < 0 \end{cases}$$

Активационная функция softmax обеспечивает сумму выходов слоя, равную единице, что позволяет трактовать выход слоя нейронной сети как вероятность событий, совокупность которых образует полную группу. Функция softmax вычисляется следующим образом:

$$f(z) = \text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}},$$

где K – число классов. На выходе функции получаются вектор, размерность которого равна числу классов, значения элементов вектора находятся в пределах [0,1], а сумма элементов вектора равна 1. Таким образом, функция softmax определяет вероятность принадлежности входного примера данных к каждому из возможных классов.

2.2.3 Обучение нейронной сети

Под обучением нейронной сети понимается нахождение ее параметров $\mathbf{W}$ и b, обеспечивающих значение целевой переменной, согласующееся с данными. В процессе обучения нейронной сети нужно вычислить функцию потерь, которая является мерой расхождения между истинным прогнозом и тем прогнозом, который выдала модель. В качестве функции потерь чаще всего используется функция перекрестной энтропии, которая определяется следующим образом:

$$L\left(y_i, \hat{y}_i\right) = -y_i \log\left(\hat{y}_i\right) - \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right)$$

В ходе обучения нейронной сети нужно подобрать параметры $\mathbf{W}$ и b таким образом, чтобы минимизировать среднее значение функции потерь:

$$\tilde{\mathbf{W}}, \tilde{b} = \arg\min_{W,b} \frac{1}{N} \sum_{i}^{N} L\left(y_i, \hat{y}_i\left(\mathbf{W}, b\right)\right),$$

где $\tilde{\mathbf{W}}$ и $\tilde{b}$ – оптимальные параметры модели. Нейронная сеть с текущими значениями параметров $\mathbf{W}$ и b получает на вход обучающий пример $\mathbf{X}_i$ и вычисляет $\hat{y}_i$. С помощью функции потерь полученный результат сравнивается со значением целевой переменной y_i, и таким образом определяется, насколько плохо или хорошо выполняется прогноз для текущего примера X_i.

Одним из основных методов нахождения оптимальных параметров модели является метод градиентного спуска. Рассмотрим работу метода более подробно. Допустим, нужно минимизировать один параметр b. Предположим, что функция потерь имеет вид, как показано на рисунке 2.7. Поиск оптимального значения параметра можно начать с какого-то случайного значения. В данном случае начальное значение параметра примем равным 1,5. Для текущей точки можно выяснить наклон функции. Зная наклон функции можно определить, в какую сторону нужно двигаться, чтобы спускаться вниз. В данном примере необходимо двигаться влево. Делаем шаг влево и получаем новую точку. Теперь нужно снова определить наклон функции и выяснить, в каком направлении двигаться дальше. После этого нужно снова сделать шаг в сторону спуска. Описанные выше действия необходимо совершать, пока не будет достигнут минимум.

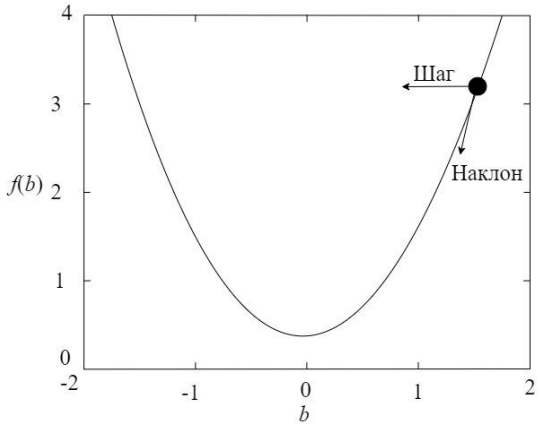


Рис. 2.7. Иллюстрация алгоритма градиентного спуска

Таким образом, алгоритм градиентного спуска состоит из следующих шагов, которые выполняются в течение нескольких итераций:

1) вычислить наклон кривой в текущей точке (если надо оптимизировать один параметр, вычислить производную, если параметров несколько – градиент $\nabla f(b^d)$, где d – номер итерации);

2) двигаться в направлении отрицательного градиента с заданным размером шагом (α), называемым также коэффициентом скорости обучения (англ. learning rate);

3) получить новую точку: $b^{d+1}=b^{d}-\alpha\nabla f(b^{d})$.

Данный алгоритм выполняется до нахождения минимума функции.

Значение параметров модели (весов и смещения) лучше выбирать случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов, и предотвращает ряд других патологических случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то модель не сможет обучиться.

При обучении методом градиентного спуска для вычисления градиентов используется полный набор данных, т.е. для выполнения градиентного спуска нужно подать на вход искусственного нейрона все примеры из обучающего множества, размер которого может быть гигантским. Поэтому для реальных задач используется не обычный, а стохастических градиентный спуск, в котором вычисление значения функции потерь и обновление весов выполняется не после прохода по всему обучающему множеству, а после каждого примера. Основное преимущество стохастического градиентного спуска состоит в том, что ошибка на каждом шаге считается быстро, веса меняются сразу же, что очень сильно ускоряет обучение.

Однако обновлять параметры модели на каждом шаге тоже может быть проблематичным, поэтому на практике обычно используется стохастический градиентный спуск по мини-батчам (англ. mini-batch) – небольшим подмножествам обучающего набора. Обычно берется несколько десятков или сотен примеров, т.е. малая часть обучающего множества, что позволяет сохранить все плюсы стохастического градиента. Один цикл прохода по всем обучающим примерам называется эпохой.

При обучении модели возникает вопрос, сколько эпох ее надо обучать. Чем большее число эпох обучается модель, тем меньше значение функции потерь на обучающем наборе данных. Однако необходимо, чтобы модель давала хорошие результаты не только на данных, которые использовались для обучения, но, что даже более важно, на новых данных. Если слишком сильно подогнать

параметры модели на обучающих данных, то может возникнуть так называемое переобучение модели, когда модель слишком хорошо настраивается на обучающие данные, но фактически теряет способность к обобщению и на новых данных работает плохо. Поэтому для того, чтобы предотвратить переобучение модели необходимо по ходу обучения проверять значение функции потерь не только на обучающем наборе данных, но и оценивать работу модели на других данных.

Таким образом, обычно выделяют три набора данных:

– обучающий набор данных - используется для обучения модели, т.е. для вычисления функции потерь и настройки параметров модели с помощью метода градиентного спуска;

– валидационный набор данных - используется для оценки модели в ходе обучения, настройки гиперпараметров (коэффициента скорости обучения и т.п.) и выбора наилучшей версии модели;

– тестовый набор данных - используется для сравнения различных моделей или различных подходов к обучению модели и для финальной оценки точности.

Использование валидационного набора позволяет, во-первых, сократить время обучения, а во-вторых, создать модель, которая будет лучше работать на реальных данных.

2.2.4 Многослойные нейронные сети

Модель искусственного нейрона хорошо работает в задачах, в которых данные могут быть разделены на классы линией (в задаче с двумерными данными), плоскостью (в задаче с трехмерными данными) или гиперплоскостью (для задач с большим числом измерений). Однако бывают ситуации, когда данные плохо разделяются линейным классификатором, в этом случае нужно использовать более сложную модель – многослойную (или как ее еще называют, глубокую) нейронную сеть. Пример такой сети с двумя скрытыми слоями показан на рисунке 2.8.

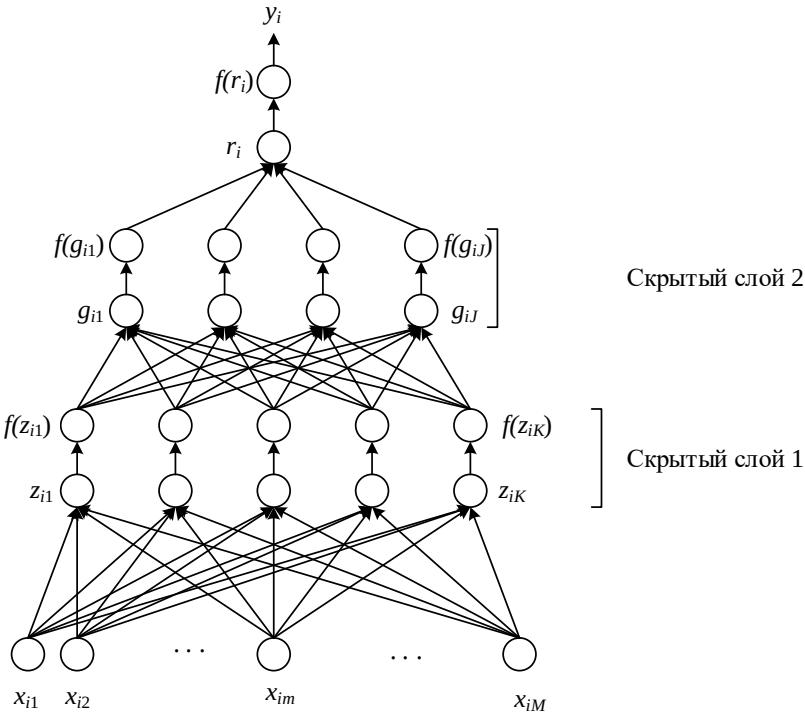


Рис. 2.8. Нейронная сеть с двумя скрытыми слоями

Первый слой сети образован путем перемножения входных данных не с одним вектором весов, как это было в модели нейрона, а с K векторами:

$$\begin{aligned} z_{i1} &= \mathbf{X}_i w_1 + b_1 \\ z_{i2} &= \mathbf{X}_i w_2 + b_2 \\ &\vdots \\ z_{iK} &= \mathbf{X}_i w_K + b_K \end{aligned}$$

Таким образом, на выходе слоя будут K выходных значений (z_{i1},z_{i2},…,z_{iK}), которые поступают на вход активационной функции. Результаты, полученные после применения активационной функции, подаются на второй слой нейронной сети. Выходные данные второго слоя (g_{i1},g_{i2},…,g_{iK}) определяются следующим образом:

$$g_{i1} = f(z_i)u_1 + c_1$$
$$g_{i2} = f(z_i)u_2 + c_2$$
$$\vdots$$
$$g_{iJ} = f(z_i)u_J + c_J,$$

где $u_1,u_2,\ldots,u_J$ – веса второго слоя нейронного сети, $c_1,c_2,\ldots,c_J$ – смещения второго слоя. Затем выходные данные второго слоя пропускаются через активационную функцию. Полученные результаты суммируются, снова прогоняются через активационную функцию и на выходе получается прогноз модели:

$$y_i = f(r_i)q + d$$

Здесь надо прояснить, зачем нужна активационная функция между скрытыми слоями нейронной сети. Если нелинейная активационная функция отсутствует, то вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением результирующего вектора на вторую весовую матрицу: $(\mathbf{XW})\mathbf{U}$. Так как умножение матриц ассоциативно, то $(\mathbf{XW})\mathbf{U}=\mathbf{X}(\mathbf{WU})$. Таким образом, двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. Поэтому, чтобы воспользоваться преимуществами многослойной нейронной сети, необходимо использовать нелинейную активационную функцию.

2.2.5 Нейронные сети с временными задержками

ИНС с временными задержками (англ. time delay neural network; TDNN) представляет собой многослойную нейронную сеть прямого распространения, узлы которой модифицированы введением временных задержек [69]. Пример узла с N задержками показан на рисунке 2.9, где $U_1\ldots U_J$ – входы узла; каждый из J входов умножается на соответствующий весовой коэффициент w; $D_1\ldots D_n$ – временные задержки, F – активационная функция [59]. Таким образом, в ИНС встраивается кратковременная память.

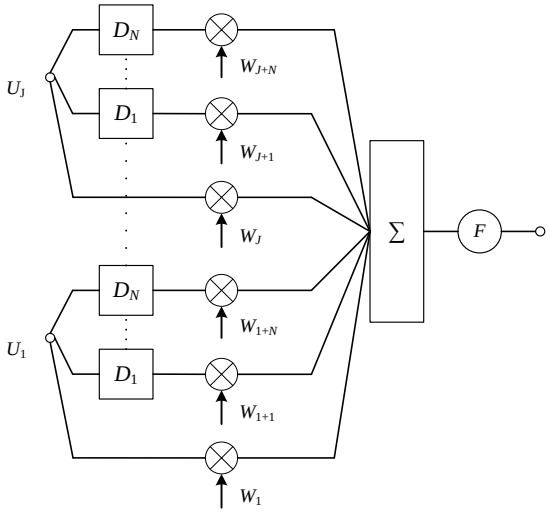


Рис. 2.9. Пример узла ИНС с временными задержками

Введение временной задержки позволяет сделать ИНС инвариантной к временным сдвигам.

2.2.6 Сверточные нейронные сети

Сверточные (англ. convolutional) нейронные сети в первую очередь применяются для обработки изображений, а также других типов данных, которые могут быть представлены в виде изображения. В частности, аудиосигнал может быть представлен в виде спектрограммы и поэтому для его обработки тоже может применяться сверточная нейронная сеть. Сверточная нейронная сеть состоит из сверточных и объединяющих (англ. pooling) слоев.

В качестве примера рассмотрим двумерную свертку. Входными данными является двумерная матрица признаков. Операция свертки состоит в следующем: ядро (англ. kernal), представляющее собой матрицу весов нейронной сети, «скользит» над матрицей входных признаков, поэлементно выполняя операцию умножения с той частью входных данных, над которой оно находится, и затем суммирует все полученные значения. Ядро повторяет эту процедуру с каждой локальной областью, над которой оно «скользит», преобразуя входную двумерную матрицу в другую двумерную матрицу признаков.

Ядро также называют фильтром. Признаки на выходе являются взвешенными суммами (где веса являются значениями самого ядра) признаков на входе. Этот процесс показан на рисунке 2.10.

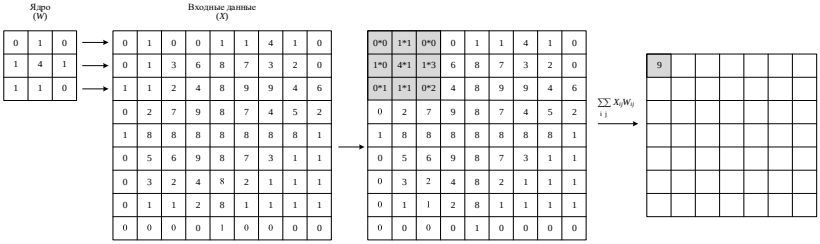


Рис. 2.10. Двумерная свертка

Размер ядра сверточной нейронной сети определяет количество признаков, которые будут объединены для получения нового признака на выходе. Ядро, показанное на рисунке 2.10, имеет размер 3x3, что является наиболее распространенным значением ядра, также часто применяют ядро размером 5x5.

Объединяющий слой располагается после сверточного слоя и активационной функции, он используется для уменьшения размера данных, которые передаются на последующие уровни сети. Применение объединяющего слоя снижает вычислительную сложность всей сети, упрощает обучение, борется с переобучением, а также обеспечивает инвариантность к расположению объекта на изображении, что означает, что нейронная сеть может идентифицировать объект на изображении, вне зависимости от его расположения. На этапе объединения также применяется фильтр, с помощью которого все значения, которые попадают в окно фильтра, объединяются в одно значение. Чаще всего такое объединение осуществляется путем нахождения максимального элемента среди элементов, попадающих в окно фильтра (англ. max pooling). Также может вычисляться среднее значение элементов (англ. average pooling). На рисунке 2.11 показан пример операции объединения путем нахождения максимума с размером окна, равным двум. Окно фильтра перемещается по карте признаков от левого верхнего угла в правый нижний угол с шагом, равным двум. Из элементов, попадающих в окно, находится максимальный.

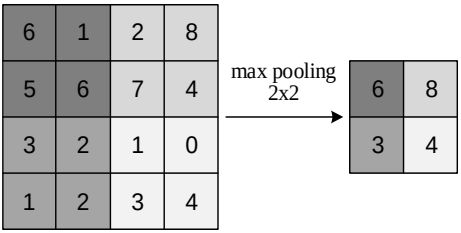


Рис. 2.11. Объединяющий слой

Следует отметить, что у объединяющего слоя нет обучаемых параметров.

После выполнения нескольких операций свертки и объединения полученные данные подаются на обычную полносвязную нейронную сеть (которая может состоять из нескольких слоев), выполняющую классификацию подаваемых на вход примеров.

2.2.7 Рекуррентные нейронные сети

Искусственные нейронные сети, описанные в предыдущих разделах, называются сетями прямого распространения. В данном разделе будут рассмотрены нейронные сети с обратными связями, которые называются рекуррентными.

Рекуррентные ИНС (РИНС) можно представить, как множество копий одной и той же сети, причем, каждая копия передает сообщение следующей копии. Пример рекуррентной сети показан на рисунке 2.12, где $\mathbf{x}_t$ – входной вектор признаков в момент времени t, $\mathbf{h}_t$ – вектор скрытого слоя, $\mathbf{y}_t$ – выходной вектор. В момент времени t на вход сети поступает входной вектор признаков $\mathbf{x}_t$ и вектор скрытого слоя, полученный на предыдущем шаге $\mathbf{h}_{t-1}$. Таким образом, скрытый слой хранит все предшествующую информацию, или как ее еще называют, историю. Предсказание выходного вектора $\mathbf{y}_t$ осуществляется не только на основе текущего вектора признаков $\mathbf{x}_t$, но и состояния скрытого слоя на предыдущем временном шаге $\mathbf{h}_{t-1}$.

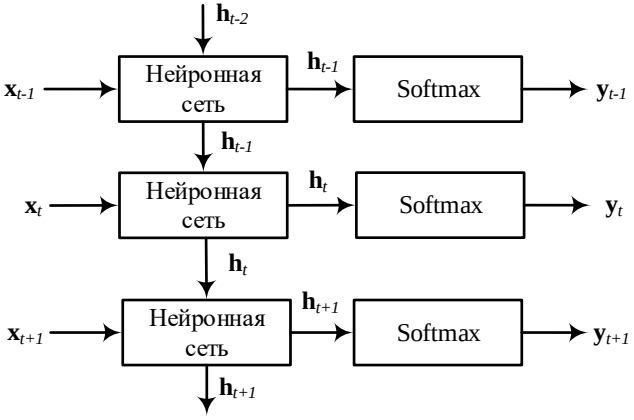


Рис. 2.12. Рекуррентная нейронная сеть, развернутая во времени

Существуют различные архитектуры РИНС: сети Хопфилда, Элмана, Джордана [68, 94], РИНС с управляемым элементами [86] и др. В настоящее время в системах автоматического распознавания речи наиболее широкое распространение получили рекуррентные нейронные сети с долгой кратковременной памятью (англ. Long Short-Term Memory; LSTM), поэтому в следующем разделе данный тип нейронных сетей будет рассмотрен более подробно.

2.2.8 Рекуррентная нейронная сеть с долгой кратковременной памятью (LSTM)

Архитектура сети LSTM показана на рисунке 2.13. В целом, архитектура сети LSTM похожа на архитектуру обычной рекуррентной сети, представленную на рисунке 2.12. Отличие состоит в том, что у сети LSTM есть ячейки памяти, выходы из которых обозначены на рисунке 2.13 как $\mathbf{c}_t$.

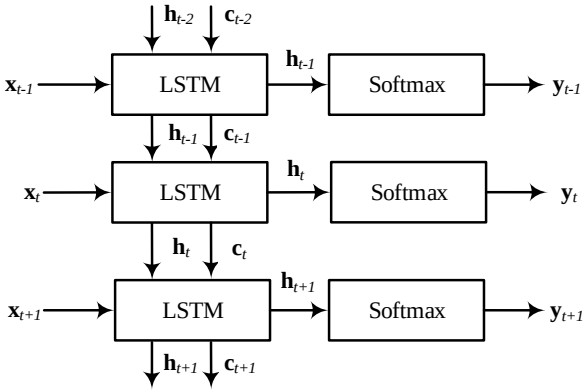


Рис. 2.13. Архитектура сети LSTM

Существенное отличие сети LSTM состоит в структуре скрытых слоев. Архитектура LSTM слоя показана на рисунке 2.14.

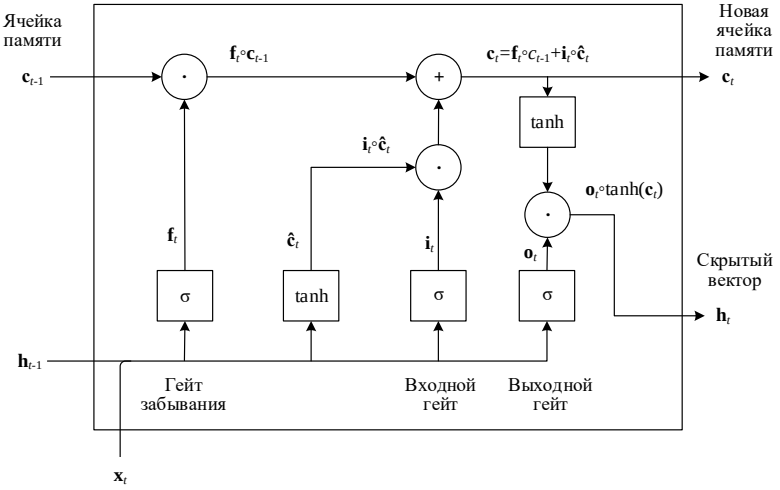


Рис. 2.14. Архитектура слоя LSTM

Слой LSTM фактически включает в себя четыре нейронных сети. Входной вектор ($\mathbf{x}_t$), а также вектор скрытого слоя в предыдущий временной шаг ($\mathbf{h}_{t-1}$) подается на вход каждой из четырех нейронных сетей. Вначале рассмотрим три из них. Эти

три нейронные сети называют фильтрами или гейтами (от англ. gates). Гейты управляют потоком информации. В стандартной архитектуре LSTM есть входной и выходной гейты и гейт забывания. Выходные векторы гейтов определяются следующим образом:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + b_i)$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + b_o)$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + b_f)$$

где $\mathbf{i}_t$, $\mathbf{o}_t$, $\mathbf{f}_t$ – векторы входного и выходного гейта и гейта забывания соответственно; $\mathbf{W}_i$, $\mathbf{W}_o$, $\mathbf{W}_f$ – матрицы весов гейта входа, выхода – и забывания соответственно, $\mathbf{U}_i$, $\mathbf{U}_o$, $\mathbf{U}_f$ – матрицы весов рекуррентных соединений, b_i, b_o, b_f – смещения входного и выходного гейта и гейта забывания соответственно, σ – сигмоидальная активационная функция. Важно отметить, что поскольку во всех трех нейронных сетях используется сигмоидальная активационная функция, значения компонентов выходных векторов $\mathbf{i}_t$, $\mathbf{o}_t$, $\mathbf{f}_t$ находятся в пределах от 0 до 1.

Четвертая нейронная сеть является ячейкой памяти. В этой нейронной сети в качестве активационной функции используется гиперболический тангенс, таким образом, компоненты выходного вектора данной нейронной сети находятся между -1 и 1. Значения выходного вектора ($\hat{\mathbf{c}}_t$) определяются следующим образом:

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + b_c),$$

где $\mathbf{W}_c$ – матрица весов, $\mathbf{U}_c$ – матрица весов рекуррентных соединений, b_c – смещение нейронной сети ячейки памяти.

Значения вектора $\hat{\mathbf{c}}_t$, являющегося обновленным представлением ячейки памяти, обрабатываются входным гейтом путем выполнения покомпонентного умножения вектора $\mathbf{i}_t$ на вектор $\hat{\mathbf{c}}_t$. Если значение компонента вектора $\mathbf{i}_t$ близко к 0, то соответствующий компонент вектора $\hat{c}_t$ не будет вносить значительный вклад в новое представление ячейки памяти. Напротив, если компонент $\mathbf{i}_t$ близок к единице, это означается, что соответствующий компонент вектора $\hat{\mathbf{c}}_t$ будет вносить значительный вклад в новую ячейку памяти $\mathbf{c}_t$.

Аналогичным образом обрабатывается вектор выхода ячейки памяти на предыдущем шаге $\mathbf{c}_{t-1}$ гейтом забывания. Выполняется покомпонентное умножение вектора $\mathbf{f}_t$ на вектор $\mathbf{c}_{t-1}$. Аналогично описанному выше, если соответствующий компонент вектора $\mathbf{c}_{t-1}$ умножается на число, близкое к 0, то фактически это означается, что этот компонент забывается.

Выход ячейки памяти $\mathbf{c}_t$ определяется следующим образом:

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \hat{\mathbf{c}}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + b_c)$$

Теперь рассмотрим, как определяется вектор скрытого состояния $\mathbf{h}_t$. Для его формирования используется выходной гейт. Аналогичным образом, как было описано выше, происходит поэлементное умножение компонентов вектора $\mathbf{o}_t$ на компоненты вектора c_t, пропущенные через гиперболический тангенс:

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

Таким образом, вектор $\mathbf{o}_t$ управляет выводом, т.е. он определяет степень, с которой каждый из компонентов вектора $\mathbf{c}_t$ влияет на состояние скрытого слоя $\mathbf{h}_t$.

Традиционная сеть LSTM может использовать только предшествующий контекст. Однако в некоторых случаях было бы полезно, кроме предшествующего контекста, учитывать также и последующий контекст. Данную задачу позволяет решить двунаправленная LSTM (англ. bidirectional LSTM; BLSTM), в которой два разных скрытых слоя берут данные в двух направлениях, а потом объединяются в один скрытый слой. Структура двунаправленной LSTM представлена на рисунке 2.15.

Модель LSTM произвела своего рода революцию в области обработки естественных языков. Она может применяться для предсказания следующего слова на основе предыдущих слов, что может быть использовано, например, для автоматического перевода с одного языка на другой, в системах распознавания речи и в чат-ботах.

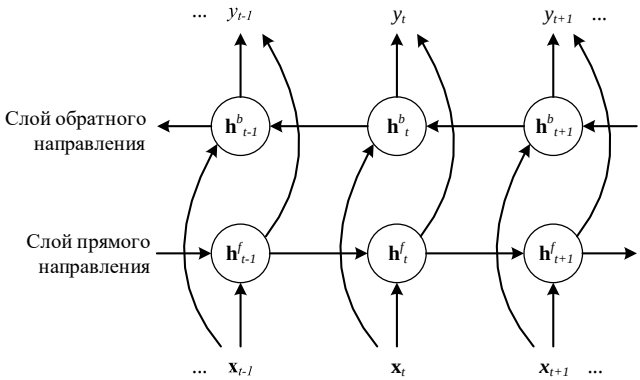


Рис. 2.15. Двунаправленная LSTM

В данном разделе были описаны основные архитектуры нейронных сетей, однако существует большое число их разновидностей. Кроме того, могут применяться различные методы, позволяющие оптимизировать процесс обучения, в частности, метод моментов Нестерова, Adagrad (от англ. adaptive gradient), Adadelta, RMSprop (от англ. root mean squares), Adam. Также существуют методы, позволяющие предотвратить переобучение нейронной сети, (например, метод dropout). Более подробное их рассмотрение выходит за рамки данного учебного пособия.

2.3 Вопросы по разделу 2

1. Что такое марковская цепь?
2. В чем состоит отличие скрытой марковской модели от простой марковской модели?
3. Что такое искусственные нейронные сети?
4. В чем состоит задача обучения искусственной нейронной сети?
5. Перечислите основные виды активационных функций.
6. Из каких шагов состоит алгоритм градиентного спуска?
7. Для чего нужен валидационный набор данных?
8. Что собой представляют сверточные нейронные сети?
9. Что такое рекуррентные нейронные сети?
10. Из чего состоит слой сети LSTM?

3 ЛЕКСИЧЕСКОЕ МОДЕЛИРОВАНИЕ РЕЧИ

3.1 Фонемный алфавит

При разработке систем автоматического распознавания речи необходимо выбрать фонемный алфавит. Чаще всего в качестве фонемного алфавита используется модифицированный и адаптированный к кириллице вариант международного фонетического алфавита SAMPA [47]. Обычно используются 46 фонем: 10 — для гласных звуков (4 безударных и 6 ударных) и 36 — для согласных (с учетом твердости и мягкости звуков). Дополнительно к обычным вариантам гласных звуков добавляются варианты с ударением, которые можно также считать аллофонами одной фонемы. Так как ударные и безударные гласные имеют значительные отличия в спектральных и временных характеристиках, то такое разделение позволяет увеличить точность моделирования речи. В таблице 3.1 представлен перечень фонем, которые используются при распознавании русской речи. В скобках дано соответствие фонем в алфавите SAMPA. В разработанном варианте знак [!] используется для обозначения ударения в слове, знак [`] — для обозначения акцентированного гласного звука (то есть второстепенного ударения в слове), знак ['] — для обозначения мягкости согласных и знак [``] до гласной означает ударность в алфавите SAMPA.

Таблица 3.1.

Перечень фонем русской речи

Фонема	Слово	Транскрипция	Фонема	Слово	Транскрипция
/а/ (/a/)	*пара*	/па!р**а**/	**/ч/ (/tS'/)**	*чай*	/**ч**а!й/
/а!/ (/''a/)	*пара*	/п**а!**ра/	**/ф/ (/f/)**	*фарс*	/**ф**а!рс/
/и/ (/i/)	*мели*	/м'э!л'**и**/	**/ф'/ (/f'/)**	*физика*	/**ф'**и!з'ика/
/и!/ (/''i/)	*мир*	/м'**и!**р/	**/в/ (/v/)**	*ваза*	/**в**а!за/
/э!/ (/''e/)	*дерево*	/д'**э!**р'ева/	**/в'/ (/v'/)**	*виза*	/**в'**и!за/
/ы/ (/1/)	*дыры*	/ды!р**ы**/	**/с/ (/s/)**	*сын*	/**с**ы!н/
/ы!/ (/''1/)	*дыры*	/д**ы!**ры/	**/с'/ (/s'/)**	*сено*	/**с'**э!на/
/у/ (/u/)	*тулуп*	/т**у**лу!п/	**/з/ (/z/)**	*запах*	/**з**а!пах/

Фонема	Слово	Транскрипция	Фонема	Слово	Транскрипция
/у!/ (/''u/)	*тулуп*	/тул**у!**п/	**/з'/ (/z'/)**	*корзина*	/карз**'**и!на/
/о!/ (/''o/)	*город*	/г**о!**рат/	**/ш/ (/S/)**	*шар*	/**ш**а!р/
/п/ (/p/)	*пыль*	/**п**ы!л'/	**/щ/ (/S':/)**	*щука*	/**щ**у!ка/
/п'/ (/p'/)	*пить*	/**п'**и!т'/	**/ж/ (/Z/)**	*жир*	/**ж**ы!р/
/б/ (/b/)	*быть*	/**б**ы!т'/	**/х/ (/x/)**	*хлеб*	/**х**л'э!п/
/б'/ (/b'/)	*бить*	/**б'**и!т'/	**/х'/ (/x'/)**	*хитрый*	/**х'**и!трый/
/т/ (/t/)	*тост*	/**т**о!ст/	**/м/ (/m/)**	*май*	/**м**а!й/
/т'/ (/t'/)	*тень*	/**т'**э!н'/	**/м'/ (/m'/)**	*мята*	/**м'**а!та/
/д/ (/d/)	*дым*	/**д**ы!м/	**/н/ (/n/)**	*найти*	/**н**айт'и!/
/д'/ (/d'/)	*день*	/**д'**э!н'/	**/н'/ (/n'/)**	*нить*	/**н'**и!т'/
/к/ (/k/)	*кот*	/**к**о!т/	**/л/ (/l/)**	*луч*	/**л**у!ч/
/к'/ (/k'/)	*кит*	/**к'**и!т/	**/л'/ (/l'/)**	*любовь*	/**л'**убо!ф'/
/г/ (/g/)	*гусь*	/**г**у!с'/	**/р/ (/r/)**	*краб*	/к**р**а!п/
/г'/ (/g'/)	*гибкий*	/**г'**и!пк'ий/	**/р'/ (/r'/)**	*резать*	/**р'**э!зат'/
/ц/ (/ts/)	*цепь*	/**ц**э!п'/	**/й/ (/j/)**	*июль*	/и**й**у!л'/

3.2 Создание базового словаря системы автоматического распознавания речи

Для функционирования системы автоматического распознавания речи необходим словарь слов с их орфографическим и фонематическим представлением, иными словами, словарь системы распознавания речи – это список слов с их транскрипциями. Пример фрагмента словаря представлен на рисунке 3.1. Важно отметить, что система распознавания речи может распознавать только те слова, которые есть в словаре. Если диктор произнесет слово, которое в словаре отсутствует, система в качестве результата распознавания выдаст другое слово, которые есть в словаре и для которого вероятность окажется выше, чем для остальных слов из словаря. Еще важно отметить, что в словаре должны быть не только начальные формы слов, но и их словоформы, т.е. не только, например, слово «стол», но и «стола», «столу», «столом» и т.д. Для системы распознавания каждая словоформа является отдельным словом.

десять	d' e! s' a t'
девяносто	d' i v' i n o! s t a
девять	d' e! v' i t'
девятнадцать	d' i v' i t n a! c a t'
девятьсот	d' i v' i c o! t
два	d v a!
двадцать	d v a! c a t'
двенадцать	d v' i n a! c a t'

Рис. 3.1. Пример словаря системы автоматического распознавания речи

Транскрипции можно сделать вручную, однако для систем с большим словарем этот процесс будет достаточно долгим и трудоемким. Поэтому обычно транскрипции создаются автоматически. Следует отметить, что существует две основные фонологические школы (Московская и Ленинградская), в которых фонетические алфавиты и правила транскрибирования несколько отличаются друг от друга [72, 81, 93, 98]. Для создания базовой системы распознавания или синтеза речи можно использовать фонетические правила транскрибирования русскоязычных текстов, описанные в [97]. Также важно отметить, что в русском языке ударение плавающее, поэтому для создания транскрипций слов необходима информация об ударениях в словах. При этом процесс простановки ударений для русского языка очень сложно автоматизировать.

При транскрибировании возможны следующие позиционные изменения классов звуков: изменения гласных в положении под ударением, изменения гласных в предударных слогах, изменения гласных в заударных слогах, позиционные изменения согласных. Далее описаны основные правила позиционных изменений звуков, которые могут быть применены в системе автоматического транскрибирования текста с учетом используемого фонетического алфавита [77].

В русском языке существует шесть позиций, в которых ударные гласные предстают в разных своих видах [96]:

– позиция в абсолютном начале слова не перед мягким согласным;

– позиция между твердыми согласными и после твердого согласного не перед согласным;

– позиция в абсолютном начале слова перед мягким согласным;

– позиция после твердого согласного перед мягким согласным;

– позиция после мягкого согласного не перед мягким согласным;

– позиция между мягкими согласными.

В таблице 3.2 представлены изменения ударных гласных. Знак «с» (consonant) принимается для обозначения любого твердого согласного, включая шипящие и /ц/, знак «с'» — для обозначения любого мягкого согласного, включая /й/, и знак «*v*» (vowel) — для обозначения любого ударного гласного.

Таблица 3.2.

Позиционные изменения гласных под ударением

Позиции ударных гласных					
v, vc	**cvc, cv**	**vc'**	**cvc'**	**c'vc, c'v**	**c'vc'**
/э/	/э/	/э/	/э/	/э/	/э/
/и/	-	/и/	–	/и/	/и/
/ы/	/ы/	–	/ы/	–	–
/а/	/а/	/а/	/а/	/а/	/а/
/о/	/о/	/о/	/о/	/о/	/о/
/у/	/у/	/у/	/у/	/у/	/у/

Позиционные изменения безударных гласных имеют место в разных позициях по отношению к ударному слогу: в предударных (иногда разделяют правила для первого предударного, а также второго и третьего предударных слогов) и заударных слогах. Позиционные изменения гласных в предударном слоге представлены в таблице 3.3, а изменения гласных в заударном слоге — в таблице 3.4.

Таблица 3.3.

Позиционные изменения гласных в предударном слоге

Гласный	Позиция				
	Начало слова	**После заднеязычных**	**После парных твердых и /ц/**	**После парных мягких и /ч/, /щ/, /й/**	**После твердых шипящих /ш/, /ж/, /ц/**
/э/	/ы/	/и/	/ы/	/и/	/ы/
/и/	/и/	/и/	–	/и/	–
/ы/	–	–	/ы/	–	/ы/
/а/	/а/	/а/	/а/	/и/	/а/
/о/	/а/	/а/	/а/	/и/	/а/
/у/	/у/	/у/	/у/	/у/	/у/

Таблица 3.4.

Позиционные изменения гласных в заударном слоге

Гласный	Позиция		
	После заднеязычных	**После парных твердых и /ц/, /ш/, /ж/**	**После парных мягких и /ч/, /щ/**
/э/	/и/	/ы/	/и/
/и/	/и/	/ы/	/и/
/ы/	–	/ы/	–
/а/	/а/	/а/	/а/
/о/	/а/	/а/	/а/
/у/	/у/	/у/	/у/

Позиционные изменения согласных фонем происходят в следующих вариантах [96]:

– в конце слова звонкие шумные оглушаются и на их месте выступают глухие шумные;

– сонорные оглушаются в конце слова после глухих шумных или перед глухими шумными;

– в положении перед глухими шумными согласными звонкие шумные согласные оглушаются, и на их месте выступают глухие шумные;

– в положении перед звонкими шумными согласными, кроме /в/, /в'/, глухие шумные озвончаются, и на их месте выступают звонкие шумные;

– в положении перед мягкими зубными /т'/, /д'/ согласные /с/, /з/ смягчаются. Перед мягкими зубными /с'/, /з'/ согласные /с/, /з/, смягчаясь, объединяются с ними в одну фонему;

– в положении перед мягкими зубными /т'/, /д'/, /с'/, /з'/ согласная /н/ произносится мягко;

– в положении перед /ч/ согласная фонема /т/ (орфографические буквы *т* и *д*), смягчаясь, объединяется с ним в фонему /ч/; в положении перед /ч/ согласная /с/ (орфогр. *с* и *з*), смягчаясь, объединяется с ним в одну фонему; сочетание букв *тщ* произносится в беглой речи как /чщ/; сочетание букв *сщ* произносится как /щ/; в положении перед /ч/, /щ/ согласная /н/ смягчается;

– в положении перед /ш/, /ж/ зубные щелевые /с/, /з/ сливаются с ними в фонему /ш/ или /ж/ соответственно;

– две одинаковые согласные, идущие подряд, заменяются одной фонемой;

– происходят изменения многобуквенных последовательностей согласных: *лнц* → /нц/, *стн* → /сн/, *здн* → /зн/, *вств* → /ств/, *фств* → /ств/, *нтг* → /нг/, *ндг* → /нг/, *ндш* → /нш/, *дст* → /цт/, *тс* → /ц/, *хг* → /г/.

Кроме того, возможно автоматически создавать транскрипции для некоторых аббревиатур. Существует три типа прочтения аббревиатур: буквенный, звуковой и буквенно-звуковой [80]. В случае буквенного типа прочтения транскрипции создаются автоматически в том случае, если транскрипция состоит только из согласных (например, СНГ — /э!сэ!нгэ!/). В этом случае буквы слова заменяются на их звуковое прочтение, а ударение ставится на последний гласный в слове. В случае звукового типа прочтения транскрипции создаются автоматически для аббревиатур вида: согласный-гласный-согласный. При этом транскрипция создается по перечисленным выше правилам транскрибирования слов (например, РАН — /ра!н/). Для аббревиатур с буквенно-звуковым типом прочтения вариант

правильного произношения не однозначен, поэтому транскрипции для них создаются вручную (например, ГУВД /гу!вэ!дэ!/ вместо /гэ!у!вэ!дэ!/ по правилам буквенного типа прочтения).

Перечисленные выше правила используются для создания базовых транскрипций слов. Однако в разговорной речи часть звуков может ассимилироваться или редуцироваться до полного исчезновения. Для учета этих явлений спонтанной речи необходимо создавать альтернативные транскрипции слов. В следующем разделе представлены основные методы, которые позволяют учитывать возможную редукцию и ассимиляцию звуков речи.

3.3 Методы моделирования вариативности произношения в разговорной речи

Одной из проблем автоматического распознавания разговорной речи является вариативность произношения слов. Одни и те же слова могут произноситься различными дикторами по-разному. Кроме того, произношение одного и того же человека может меняться в зависимости от стиля и темпа речи. Например, слово «шестьдесят», которое имеет базовую транскрипцию /ш ы з' д' и с' á т/, соответствующую каноническому произношению, в разговорной речи часто произносится как /ш ы с' á т/ или даже /ш с' á т/. Это приводит к снижению точности распознавания речи. Качественная и количественная редукция гласных, ослабление согласных, выпадение согласных, уменьшение степени контрастности между гласными и согласными в пределах слога являются основными особенностями спонтанной речи [92]. Поэтому фонетическое представление произнесенных слов часто не совпадает с транскрипциями, сделанными для изолированных слов по фонетическим правилам русского языка.

В современных системах распознавания разговорной речи для учета вариативности произношения помимо базовых транскрипций, которые создаются по фонетическим правилам, в словарь включают список альтернативных транскрипций. От метода, выбранного для создания альтернативных транскрипций, будет зависеть, насколько точно они отражают вариативность разговорной речи. Существует два основных подхода к описанию

вариативности произношения [1]: методы, основанные на знаниях, и методы, основанные на данных (см. рисунок 3.2). Рассмотрим эти методы более подробно.

Рис. 3.2. Методы учета вариативности произношения

В методах, основанных на знаниях, вариативность произношения определяется путем анализа существующих фонетических и лингвистических знаний, сформулированных экспериментальной фонетикой в ходе анализа речевых данных и акустико-артикуляторных характеристик фонем. Главная проблема этих методов заключается в том, что они описывают только часто возникающие отклонения в речи, при этом учитывается достаточно узкий фонетический контекст. Генерация транскрипций производится путем применения всех возможных комбинаций правил к базовым транскрипциям, при этом обычно не учитывается частота встречаемости правил, а также их комбинаций. Поэтому в зависимости от количества используемых правил синтезируется слишком много или слишком мало вариантов произношения.

В методах, основанных на данных, альтернативные транскрипции создаются в процессе анализа обучающих многодикторных корпусов спонтанной речи. Собранные реальные транскрипции слов могут описать только те отклонения, которые встретились в данном корпусе. Поэтому полнота альтернативных транскрипций будет напрямую зависеть от размера и представительности речевого корпуса. В отличие от методов, основанных на знаниях, здесь по обучающему корпусу можно посчитать вероятности появления каждой альтернативной транскрипции и частоту возникновения каждого типа отклонения. Тем не менее, если корпус недостаточно представителен, то полученные альтернативные транскрипции, а также их вероятности могут быть характерны для некоторого частного случая и не смогут описать все возможные отклонения, возникающие в спонтанной речи.

В обоих вариантах учета вариативности произношения используют прямое и косвенное моделирование. В методах, основанных на знаниях, прямое моделирование осуществляется путем последовательного анализа каждой базовой транскрипции и добавления альтернативных вариантов произношения с учетом знаний эксперта. При косвенном моделировании используется некоторый набор правил редукции и ассимиляции, которые применяются для синтеза альтернативных транскрипций по имеющемуся списку базовых транскрипций.

При прямом моделировании в методах, основанных на данных, в качестве альтернативных транскрипций используются только часто встречающиеся в обучающем корпусе варианты произношения слов. При косвенном моделировании выявляются наиболее характерные изменения в произношении одинаковых цепочек фонем в различных словах, то есть по речевому корпусу определяют правила наиболее типичных изменений на уровне фонем.

Для обоих подходов при выборе метода косвенного моделирования необходимо правильно обобщать (формализовать) наблюдаемую вариативность. Если некоторая вариативность появляется в сильно различающихся контекстах, обобщение может быть неправомерным. Так, наблюдаемая вариативность в служебных словах не обязательно подходит для

ключевых слов, даже если фонетический контекст является тем же.

Чтобы избежать появления некорректных правил в подходе, основанном на данных, анализируются два корпуса речи: обучающий и тестовый. В ходе косвенного моделирования вариативности транскрипций выбираются только те правила изменения произношения, которые были выявлены в обучающем корпусе и подтвердились при анализе тестового корпуса речи.

Существует ряд причин, по которым применение косвенного моделирования в подходе, основанном на данных, является более перспективным. Во-первых, списки слов, использованных в обучающем и тестовом корпусах, могут отличаться. Поэтому альтернативные транскрипции для слов, которые присутствуют в обучающем корпусе, но отсутствуют в тестовом корпусе, будут отклонены. Использование правил позволяет описать наиболее характерные изменения, которые наблюдаются в ряде слов, что обеспечивает сохранение правил в процессе их проверки на тестовом корпусе. Во-вторых, правила строятся для цепочек фонем, а не полных транскрипций, и поэтому одно и то же правило может выполняться в различных словах. Так как статистика по встретившимся правилам будет более представительной, чем статистика по альтернативным транскрипциям, то и достоверность правил будет более высокой. Наконец, прямое моделирование межсловных явлений не представляется возможным, поскольку это требует полного перебора всех возможных комбинаций транскрипций слов. В случае косвенного моделирования возможные изменения на стыке слов могут быть заданы с помощью правил, которые применяются для любых комбинаций слов. В некоторых случаях для часто употребляемых пар знаменательных и служебных слов создаются базовые транскрипции, в которых уже учтены межсловные изменения на стыке служебного и знаменательного слова [14].

Зачастую используют комбинацию методов, основанных на знаниях и данных. Например, используя знания о явлениях редукции и ассимиляции, составляется некоторый набор правил, по которому синтезируется словарь, включающий альтернативные транскрипции. Затем по размеченному вручную речевому корпусу проверяется, какие из альтернативных

транскрипций действительно существуют, и оцениваются вероятности соответствующих правил редукции и ассимиляции.

При создании словаря альтернативных транскрипций обычно используется несколько итераций. При нахождении новой транскрипции требуется переобучение акустических моделей путем повторного транскрибирования обучающего набора с использованием новой транскрипции. Обычно это приводит к увеличению точности распознавания. Последующая итерация с использованием переобученных моделей для новых транскрипций позволяет выявить новый набор транскрипций, что может дать еще большее увеличение производительности. Моделирование вариативности произношения на уровне лексических моделей уменьшает необходимость моделирования вариативности в акустических моделях. Переобучение акустических моделей после модификации словаря транскрипций дает лучший результат распознавания из-за того, что модели лучше отражают обучающие данные.

3.4 Создание словаря, моделирующего вариативность произношения

В данном разделе описан процесс создания расширенного словаря, моделирующего вариативность произношения на примере применения комбинированного метода создания транскрипций. Суть метода состоит в том, что вначале путем применения правил редукции и ассимиляции звуков речи по списку базовых транскрипций создается словарь, включающий альтернативные транскрипции. Затем по размеченному вручную речевому корпусу проверяется, какие из альтернативных транскрипций действительно существуют.

Формализованные и адаптированные для задачи распознавания речи правила создания базовых и альтернативных транскрипций представлены в работе [77]. Ниже представлены примеры расширенных правил транскрибирования:

– безударные гласные редуцируются до полного исчезновения, если они находятся между одинаковыми согласными (балалайка /балала!йка/ → /балла!йка/);

– согласная фонема /й/ в конце слова редуцируется до полного исчезновения, если ей предшествует безударная гласная, а следующее слово начинается с любой фонемы, кроме ударной

гласной (*драгоценный камень* /драгацэ!ный ка!м'ин'/ → / драгацэ!ны ка!м'ин'/);

– первая в слове гласная /и/ после всех твердых согласных переходит в фонему /ы/ (*фильм интересный* /ф'и!л'м ин'т'ир'э!сный/ → /ф'и!л'м ын'т'ир'э!сный/).

Схема комбинированного метода транскрибирования текстов показана на рисунке 3.3.

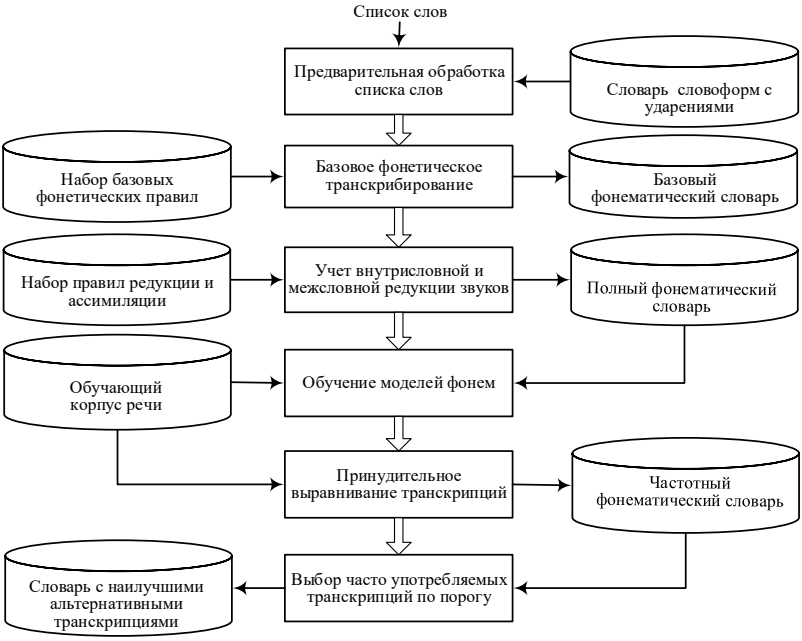


Рис. 3.3. Схема метода создания фонематического словаря системы распознавания речи

Вначале создаются базовые транскрипции словоформ. При этом к списку слов последовательно применяются базовые фонетические правила транскрибирования. Затем полученный список базовых транскрипций поступает на уровень учета внутрисловной и межсловной редукции и ассимиляции, где для каждой базовой транскрипции слова определяется, какие фонемы могут быть подвержены изменению. Если таких потенциальных фонем в слове больше одной, то производится генерация всех возможных вариантов произношения слова. Полученный таким

образом набор транскрипций теоретически должен содержать все варианты произношений, которые могут возникать в разговорной речи различных людей.

Обучение моделей фонем (т.е. скрытых марковских моделей) осуществляется с использованием полного фонематического словаря. Для выбора из множества альтернативных транскрипций и сокращения избыточности словаря осуществляется их принудительное выравнивание (англ. forced alignment) [28], при котором распознаватель выбирает из списка альтернативных транскрипций наиболее подходящую речевому сигналу и сегментирует сигнал на фонемы с их временными метками. В этом случае выбор транскрипции происходит только между альтернативными транскрипциями одного и того же слова, а не между транскрипциями разных слов.

Основой процедуры принудительного выравнивания является алгоритм Витерби (представлен в главе 4), который находит оптимальную последовательность состояний СММ на основе максимальной акустической вероятности соответствия модели сигналу. Суть классического алгоритма Витерби заключается в задании начальных параметров модели с последующим чередованием фаз оценки и максимизации параметров по критерию максимального правдоподобия. Для каждого выравнивания алгоритм Витерби вычисляет вероятность того, что фонематическая транскрипция и речевой сигнал подходят друг другу. Наибольшие вероятности при выравнивании транскрипций каждого слова позволяют выбрать оптимальные варианты транскрипций. В результате выполнения принудительного выравнивания выбирается транскрипция, наиболее оптимально подходящая определенному участку речевого сигнала. Транскрипции, которые ни разу не выбрались при принудительном выравнивании, исключаются из словаря, и таким образом, создается сокращенный словарь транскрипций. Однако этот сокращенный словарь также еще является избыточным и содержит редкие варианты произношения, что приводит к увеличению акустической и лексической неоднозначности. Поэтому для уменьшения избыточности словаря производится анализ того, насколько часто каждая альтернативная транскрипция выбиралась в ходе обучения, и создается частотный словарь транскрипций. Таким образом, в

итоговый расширенный словарь добавляются только те транскрипции, относительная частота появления которых (то есть отношение числа появлений транскрипции в речевом корпусе к числу появлений слова в орфографическом представлении обучающего корпуса) выше определенного задаваемого эмпирически порога. В результате создается расширенный (относительно базового) словарь фонематических транскрипций, содержащий наилучшие транскрипции для каждого слова, встретившиеся в обучающем речевом корпусе.

На следующем этапе разработки системы автоматического распознавания речи необходимо создать акустические модели для фонетических единиц речи (фонем). Процесс акустического моделирования речи описан в следующем разделе.

3.5 Вопросы по разделу 3

1. В чем состоит проблема вариативности речи для автоматических систем?
2. Для чего применяются фонематические транскрипции в системах распознавания речи?
3. Как в системе распознавания речи используется словарь произношения слов?
4. Какие существуют методы моделирования вариативности произношения?
5. В чем состоит суть моделирования вариативности произношения в методах, основанных на знаниях?
6. В чем состоит суть моделирования вариативности произношения в методах, основанных на данных?
7. Как осуществляется прямое и косвенное моделирование вариативности речи?
8. В чем состоит суть комбинированного метода моделирования вариативности произношения слов?
9. Какая информация содержится в расширенном словаре фонематических транскрипций слов?
10. Что понимается под принудительным выравниванием транскрипций в методах, основанных на данных, и какой алгоритм для этого используется?

4 АКУСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РЕЧИ

4.1 Параметрическое представление сигнала

Важным вопросом, с которым в первую очередь сталкивается разработчик речевых технологий – это разработка оптимального метода параметрического представления сигнала, который позволил бы достаточно хорошо различать звуки и слова речи и в то же время обеспечить инвариантность к особенностям произношения конкретного диктора и к изменениям акустической обстановки [90].

В системах распознавания речевой сигнал делится на короткие сегменты, и каждый сегмент преобразуется в вектор признаков, в результате входной сигнал представляется последовательностью векторов признаков. Процесс вычисления векторов признаков называется извлечением признаков или параметрическим представлением. Речь можно представить и моделировать как стохастический (случайный) процесс, который создает последовательность независимых векторов признаков. Стохастический процесс — это процесс, течение которого может быть различным в зависимости от случая и для которого определена вероятность того или иного его течения [89]. Целью процесса извлечения признаков является преобразование входного сигнала в некоторую компактную форму параметрического представления.

Микрофоны улавливают давление звуковых волн, распространяющихся в воздухе, и преобразуют их в электрические сигналы. Перед процессом извлечения признаков эти сигналы должны быть предварительно оцифрованы. Стандартный аналого-цифровой преобразователь (АЦП) производит дискретизацию непрерывного сигнала по времени и квантование амплитуды сигнала в необходимом диапазоне значений. Первым шагом является преобразование непрерывного аналогового сигнала в дискретный сигнал с помощью дискретизации временной области. Величина сигнала измеряется в определенные моменты времени с периодом дискретизации t_0. Частота дискретизации определяется как $f_0 = \frac{1}{t_0}$. На втором шаге непрерывные значения амплитуды записанного сигнала

дискретизируются для того, чтобы можно было их представить в машинном коде с конечным числом разрядов. Этот шаг называется квантованием. Большинство современных распознавателей используют 16 бит для хранения одного АЦП-отсчета (одного значения сигнала). Последний шаг предварительной обработки – фильтр предыскажения, где сигнал подвергается свертке с помощью рекурсивного фильтра высоких частот первого порядка. Цель такого предыскажения заключается в том, чтобы подчеркнуть высокочастотные компоненты звуков речи, которые обычно имеют уменьшенную амплитуду вследствие ослабления высоких частот в речевом сигнале, в частности, для звонких согласных звуков речи [78].

Вектора признаков обычно вычисляются для коротких сегментов сигнала (кратковременный анализ) с использованием допущения, что речь может рассматриваться как стационарная на этих коротких интервалах. Для более точного описания сигнала речевые сегменты берутся с перекрытием. Процесс создания речевых сегментов выполняется с помощью метода окна, то есть путем перемножения сигнала с некоторой функцией окна для того, чтобы разрывы на границах окна были ослаблены. Обычно для этих целей используется окно Хэмминга, в этом случае функция окна принимает вид:

$$w(k) = \begin{cases} 0,54 - 0,46\cos\left(\dfrac{k2\pi}{K-1}\right), \text{ при } k = 0,1,...,K-1 \\ 0, \text{ при } k \neq 0,1,...,K-1 \end{cases},$$

где K – ширина окна.

В программной реализации вычисление предыскажающего фильтра и применение окна Хэмминга выполняют одновременно в одной рекурсивной процедуре.

На рисунке 4.1 показано окно Хэмминга со структурой из 100 отсчетов. Окно имеет длительность 25 мс (500 отсчетов при 20 кГц) с 15 мс (300 отсчетов) перекрытием. Следующим шагом выделения признаков является преобразование каждого фрейма из временной области в частотную область (вычисления спектра сигнала) путем использования дискретного преобразования Фурье. Этот шаг обычно выполняется как быстрое

преобразование Фурье, которое является эффективной реализацией дискретного преобразования Фурье. Для того чтобы быстрое преобразование Фурье было эффективным, преобразуемая длина должна иметь степень 2. Для сегментов из 500 отсчетов преобразуемая длина должна быть 512 с дополнением нулей к концу сегмента, чтобы иметь 512 отсчетов. Вычисляется квадрат значения быстрого преобразования Фурье, поскольку необходимо вычислить значение энергии частоты.

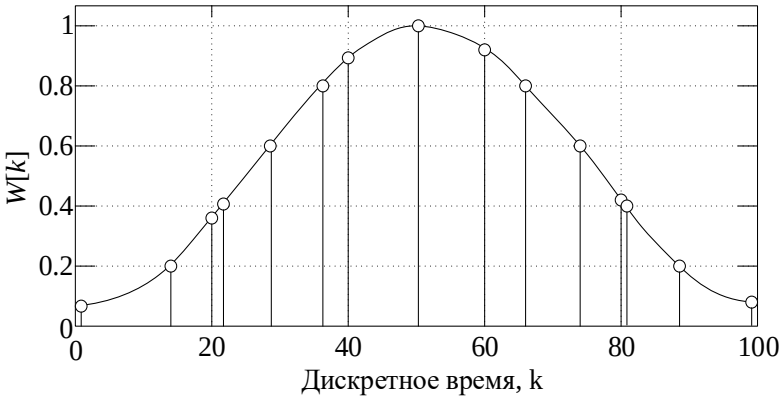


Рис. 4.1. Пример окна Хэмминга для обработки сегмента речи

Затем для каждого фрейма выполняется анализ с помощью гребенки мел-фильтров, для того чтобы более точно моделировать восприятие звуковой энергии слуховой системой человека. Мел-фильтры основаны на мел-шкале, которая является логарифмической шкалой, аналогичной слуховому восприятию человека. Мел–шкала определяется по следующей формуле относительно шкалы частот f, измеряемой в герцах:

$$f_{mel} = 2595 \cdot \lg\left(1 + \frac{f}{700}\right) \qquad (2)$$

Гребенка мел-фильтров выполняется с перекрывающимися треугольными весовыми функциями. Эти треугольные весовые функции покрывают одинаковые полосы пропускания в мел-шкале с перекрытием 50 % (то есть мел-фильтр покрывает частоты между центральными частотами соседних фильтров).

Поэтому они представляются равными областями на мел-шкале, в то время как их ширина в частотной области возрастает логарифмически с частотой. Фильтры могут иметь одинаковую ширину на мел-шкале и затем быть отображены в частотной области с помощью формулы 2. На рисунке 4.2 показан пример треугольных весовых функций из набора мел-фильтров с 12 полосами на частотной оси.

При автоматическом анализе необходимо выполнить оценку энергии спектра в каждой мел-полосе (фильтре). Вектор, составленный из энергии всех элементов разрешения по частоте в каждом мел-фильтре, представляет собой мел-спектр.

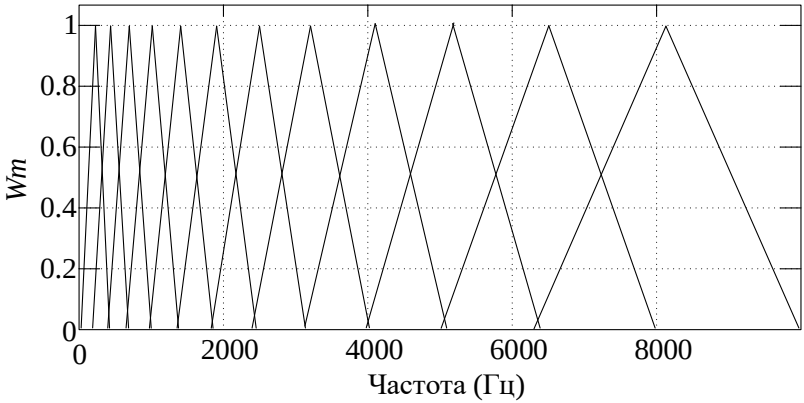


Рис. 4.2. Гребенка 12-полосных мел-частотных фильтров

Из-за перекрытия треугольных весовых функций мел-фильтров смежные компоненты вектора мел-спектра взаимно коррелируют друг с другом. С применением дискретного косинусного преобразования на логарифмическом мел-спектре взаимная корреляция между смежными компонентами сильно уменьшается. На выходе цифровой обработки сигналов формируются так называемые мел-частотные кепстральные коэффициенты.

На рисунке 4.3 показана схема процесса вычисления мел-спектра, логарифмического мел-спектра и мел-частотных кепстральных коэффициентов. Мел-частотные кепстральные коэффициенты, в основном, используют набор векторов признаков при распознавании речи, так как они улучшают работу по отношению к большинству других параметров. Многие

системы распознавания речи объединяют производные коэффициентов первого и второго порядков для того, чтобы уловить спектральные изменения и улучшить работу систем автоматической обработки и распознавания речи.

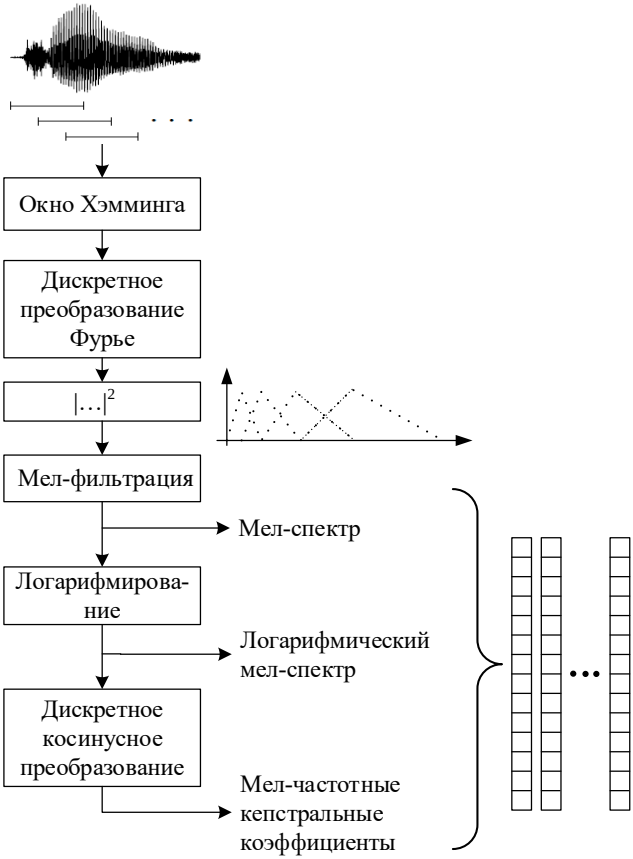


Рис. 4.3. Схема процесса извлечения мел-частотных кепстральных коэффициентов из сегмента речи

После извлечения из речевого сигнала векторов-признаков можно приступать к обучению акустических моделей фонем. Далее описаны два основных способа создания акустических моделей, применяемых в настоящее время, – на основе СММ и на основе ИНС.

4.2 Акустическое моделирование речи на основе скрытых марковских моделей

СММ представляет звук как последовательность дискретных стационарных состояний с мгновенными переходами между ними. Каждая фонема описывается своей СММ из N состояний (рисунок 4.4). Состояния цепи соответствуют измеряемым векторам, а переходы возможны только слева направо. Дуги, замкнутые на себя, моделируют временную изменчивость фонем. Во время обучения системы итерационным путем определяются вероятности всех переходов и вероятности того, что определенный набор акустических свойств (вектор) может наблюдаться в каждом из состояний цепи. Каждое слово, в свою очередь, представляется в словаре системы как СММ из нескольких фонем, а каждая фраза – как СММ из нескольких слов.

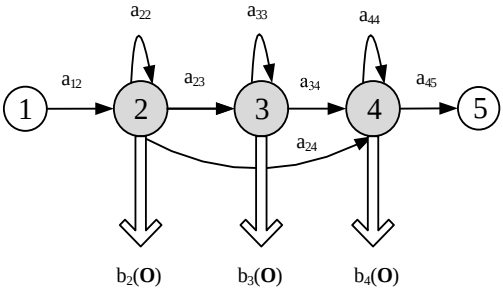


Рис. 4.4. СММ для моделирования звука речи (фонемы)

Существует два режима работы с моделью – обучение и распознавание. При обучении вероятности вычисляются по известным произносимым словам. СММ при обучении позволяет с помощью вероятности отразить те закономерности речи, которые не наблюдаются в сигнале. При распознавании строятся все возможные переходы в зависимости от сигнала, и определяется вероятность того, что в конце мы окажемся в конкретном состоянии.

Для полного определения дискретной СММ λ некоторой речевой единицы необходимо задать следующие параметры:

1) N – число состояний в модели; хотя состояния скрыты от наблюдателя, в некоторых практических задачах состояниям или множествам состояний модели приписывается некий физический

смысл (например, фонемы или буквы); для обозначения множества состояний модели используется запись $S=\{s_1,s_2,\ldots,s_N\}$, а состояние модели в момент t обозначается q_t;

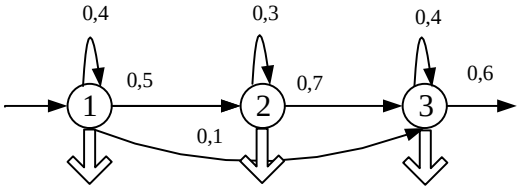
2) M – число различных символов наблюдения, которые могут порождаться моделью, то есть размер дискретного алфавита; символы наблюдения соответствуют физическому выходу моделируемой системы; множество наблюдаемых символов обозначается как $V=\{v_1,v_2,\ldots,v_M\}$;

3) распределение вероятностей переходов между состояниями (или матрица переходных вероятностей) $\mathbf{A}=\{a_{ij}\}$, где $a_{ij}=P[q_{t+1}=s_i|q_t=s_j]$;

4) распределение вероятностей появления символов наблюдения в состоянии j, $B=\{b_j(\mathbf{o})\}$, где $b_j(\mathbf{o})=P[v_{ot}=s_i|q_t=s_j]$;

5) начальное распределение вероятностей состояний $\pi=\{\pi_i\}$, $\pi_i=P[q_1=s_i]$.

На рисунке 4.5 дан пример задания топологии и параметров дискретной СММ.



0	0,03
1	0,06
2	0,07
3	0,09
4	0,15
5	0,25
6	0,20
7	0,07
8	0,05
9	0,03
o	$b_1(\mathbf{o})$

0	0,02
1	0,03
2	0,05
3	0,07
4	0,08
5	0,10
6	0,30
7	0,20
8	0,09
9	0,06
o	$b_2(\mathbf{o})$

0	0,05
1	0,09
2	0,10
3	0,20
4	0,15
5	0,12
6	0,10
7	0,08
8	0,07
9	0,04
o	$b_3(\mathbf{o})$

Рис. 4.5. Пример задания скрытой марковской модели

Проблема, возникающая при использовании дискретных СММ, заключается в том, что в большинстве практических задач наблюдения являются непрерывными сигналами (или векторами) и их квантование с помощью кодовых книг размерностью M

может иногда приводить к искажениям исходного сигнала. Поэтому часто для распознавания речи используют СММ с непрерывными плотностями вероятностей для векторов наблюдений. В таких моделях плотность вероятности появления векторов наблюдений описывается следующим образом:

$$b_i(\mathbf{O}) = \sum_{m=1}^{M} C_{jm} \vartheta\left[\mathbf{O}, \mu_{jm}, U_{jm}\right]$$

где **O** – моделируемый вектор наблюдений, C_{jm} – весовой коэффициент m-й компоненты в состоянии j и ϑ - произвольная логарифмически-вогнутая или эллиптически-симметричная плотность вероятности с вектором средних значений (математическим ожиданием) μ_{jm} и ковариационной матрицей (дисперсией или отклонением от матожидания) $\mathbf{U}_{jm}$ для m-составляющей в состоянии j. Как правило, в качестве плотности вероятности используется гауссовская плотность. Плотности такого вида часто используются на практике, поскольку позволяют с любой точностью аппроксимировать произвольную непрерывную функцию, содержащую конечное число компонент. На рисунке 4.6 показан пример из трех одномерных гауссовских плотностей вероятности (показаны сплошными линиями) и их смеси (показано пунктирной линией).

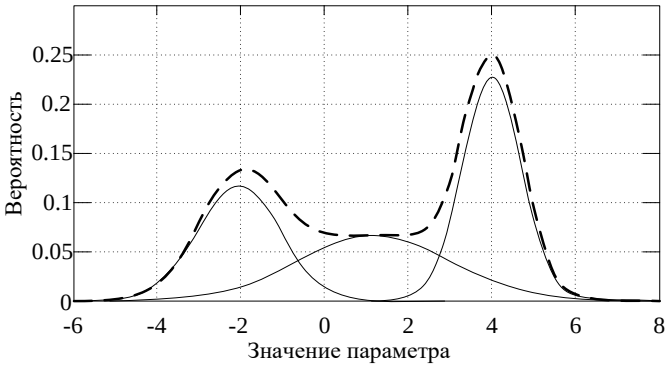


Рис. 4.6. Смесь трех одномерных гауссовских плотностей вероятности

Существуют три основные задачи, связанные с использованием СММ для распознавания речи [3]:

Задача 1 (оценка вероятности). Пусть задана последовательность наблюдений $\mathbf{O}=(o_1,o_2,\ldots,o_T)$ и модель λ. Как эффективно вычислить величину $P(\mathbf{O}|\lambda)$, т. е. вероятность появлений этой последовательности наблюдений для данной модели? Существует несколько способов оценки правдоподобия. Наиболее широко применяют алгоритмы прямого-обратного хода (forward-backward), а также лучевые алгоритмы [44].

Задача 2 (декодирование). Пусть заданы последовательность наблюдений $\mathbf{O}=(o_1,o_2,\ldots,o_T)$ и модель λ. Как выбрать последовательность состояний $Q=(q_1,q_2,\ldots,q_T)$, которая в некотором значимом смысле будет оптимальной (например, наилучшим образом соответствует имеющейся последовательности наблюдений)? Для решения этой задачи применяют алгоритм Витерби [44, 58].

Задача 3 (обучение). Каким образом нужно подстроить параметры модели λ, для того чтобы максимизировать $P(\mathbf{O}|\lambda)$? Задача обучения СММ является крайне важной и в то же время наиболее трудной задачей.

Цель обучения акустических моделей состоит в том, чтобы по заданной последовательности наблюдений определить метод такой подстройки параметров модели, чтобы для полученной модифицированной модели вероятность появления этой последовательности была максимальной. Не существует известного аналитического выражения для настройки параметров такой модели. Кроме того, на практике, располагая некоторой последовательностью наблюдений в качестве обучающих данных, нельзя указать оптимальный способ оценки параметров. Тем не менее, используя итеративные процедуры, например метод Баума-Уэлча, EM (expectation maximization) метод или градиентные методы [4, 44], можно выбрать параметры модели таким образом, чтобы локально максимизировать вероятность $P(\mathbf{O}|\lambda)$.

Если итеративно повторять процедуру переоценки параметров, используя на каждом новом шаге значения параметров модели, полученные на предыдущем шаге, то будем последовательно получать модели, для которых вероятность появления последовательности наблюдений **O** будет

увеличиваться. Процедура продолжается до тех пор, пока не будет достигнута некоторая предельная точка (например, по критерию максимума правдоподобия СММ).

Согласно теории, описанной в [44], процедура переоценки должна давать значения параметров СММ, которые соответствуют локальному максимуму функции правдоподобия. И при этом крайне важным является вопрос, как выбирать начальные значения параметров заданной модели, для того чтобы локальный максимум оказался глобальным максимумом функции правдоподобия.

Исследования показывают [44], что либо случайные (подверженные стохастичности и ограничениям ненулевых значений), либо однородные начальные оценки параметров π и $\mathbf{A}$ почти во всех случаях позволяют получать вполне приемлемые повторные оценки для этих параметров. Что же касается параметра B, то хорошие начальные оценки являются полезными в случае дискретных символов и необходимы в случае непрерывного распределения. Такие начальные оценки могут получаться несколькими различными способами, включая ручную сегментацию последовательностей наблюдений на состояния с усреднением числа наблюдений в состояниях, сегментацию наблюдений по методу максимального правдоподобия с усреднением, сегментацию с использованием метода k-средних [25] и т.д.

После инициализации модели множество обучающих последовательностей наблюдений разбивается на состояния в соответствии с используемой моделью λ. Такое разбиение достигается посредством нахождения оптимальной последовательности состояний с помощью алгоритма Витерби и последующего поиска в обратном направлении вдоль оптимального пути. Результатом разбиения на состояния каждой обучающей последовательности также является вероятностная оценка принадлежности множества наблюдений конкретной модели. Обновленная модель λ' получается на основе вычисленных параметров модели, а переоценка всех параметров этой модели выполняется с помощью процедуры повторного оценивания. Результирующая модель сравнивается с предыдущей моделью посредством вычисления меры отклонения, которая отражает статистическое сходство этих

моделей. Если эта мера отклонения моделей превышает порог, старая модель λ заменяется новой моделью λ′ (для которой выполняется процедура переоценки), и полностью повторяется цикл обучения. Если же мера отклонения не превышает данного порога, то полагается, что модель сходится, и сохраняются параметры последней модели.

На этапе автоматического распознавания речи строятся всевозможные переходы по состояниям СММ и определяется вероятность того, что в конце мы окажемся в конечном состоянии, используя алгоритм прямого-обратного хода или алгоритм Витерби. Алгоритм Витерби применяют для распознавания как изолированной, так и слитной речи. Он состоит из прямого и обратного проходов и реализуется следующим образом. Для начала необходимо ввести следующую переменную [44]:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P\left[q_1 q_2 \ldots q_{t-1}, q_t = i, o_1 o_2 \ldots o_t \mid \lambda\right],$$

имеющую смысл максимальной вероятности того, что при заданных наблюдениях до момента времени t последовательность состояний завершится в момент времени t в состоянии i. Также введем переменную $\psi_t(j)$ для хранения аргументов, максимизирующих $\delta_t(i)$. Алгоритм состоит из 4 шагов:

1) инициализация:

$$\delta_1(j) = \pi_i b_i(o_1), 1 \le i \le N,$$
$$\psi_1(i) = 0,$$

2) индуктивный переход:

$$\delta_t(j) = \max_{1 \le i \le N}\left[\delta_{t-1}(i)\alpha_{ij}\right] b_j(o_t), 1 \le j \le N, 2 \le t \le T$$
$$\psi_t(j) = \arg\max_{1 \le i \le N}\left[\delta_{t-1}(i)\alpha_{ij}\right],$$

3) остановка

$$P^* = \max_{1 \le i \le N} \left[\delta_T (i) \right],$$

определяющая максимальную вероятность наблюдения последовательности O, которая достигается при прохождении некой оптимальной последовательности состояний $Q^*=(q_1^*,\ldots,q_T^*)$, для которой к настоящему моменту известно только последнее состояние:

$$q_T^* = \arg \max_{1 \le i \le N} \left[\delta_T (i) \right],$$

4) восстановление оптимальной последовательности состояний (обратный проход):

$$q_T^* = \psi_{t+1} \left(q_{t+1}^* \right), t = T-1, T-2, ..., 1$$

Результатом работы алгоритма является наибольшая вероятность появления распознаваемой последовательности наблюдений для заданной СММ, то есть степень близости слова (или цепочки слов), задаваемого данной моделью, к распознаваемому сигналу. Причем с помощью алгоритма Витерби можно как вычислить вероятность принадлежности последовательности наблюдений некоторой СММ, так и узнать оптимальную последовательность пройденных состояний модели.

Использование марковского моделирования для распознавания изолированных слов можно упрощенно разделить на два этапа: первый этап – создание СММ для каждого слова из словаря с объемом V, а также оптимизация их параметров, и второй этап - распознавание. Для каждого неизвестного слова, подлежащего распознаванию, применяется обработка, показанная на рисунке 4.7 где определяется последовательность наблюдений $\mathbf{O}=(o_1,o_2,\ldots,o_n)$ путем анализа речевого сигнала, затем производится вычисление вероятностей правдоподобия всех возможных гипотез $P(\mathbf{O}|\lambda^v)$, где $v \in [1,V]$. Модель, вероятность правдоподобия которой наибольшая, считается

оптимальной гипотезой сказанного слова, то есть индекс распознанного слова v^* вычисляется следующим образом:

$$v^* = \arg\max_{v\in[1,V]}\left[P\left(\mathbf{O}\mid\lambda^v\right)\right]$$

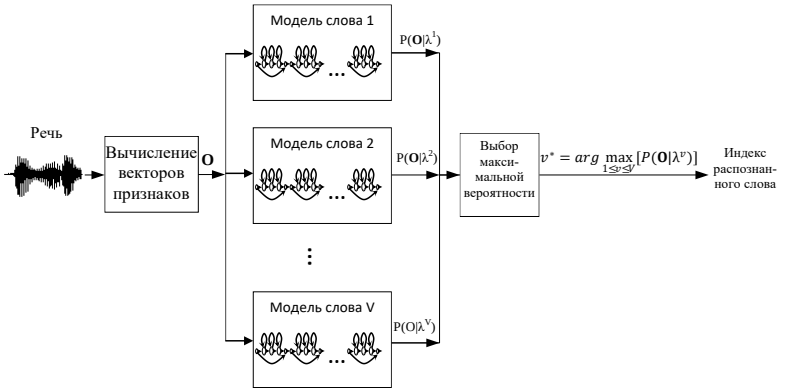


Рис. 4.7. Распознавание изолированных слов на основе СММ

Разработка систем распознавания, основанных на СММ, требует значительных объемов акустических данных, позволяющих создавать необходимые стохастические модели. При создании речевых корпусов учитывается множество факторов, таких как характеристика диктора (национальность, пол, возраст), канал передачи данных (микрофон, телефон), уровень шума. Эти базы данных должны также содержать фонематические транскрипции, разметку акустического сигнала по фонемам, слогам, словам, фразам.

4.3 Метод распознавания слитной речи

Для работы со слитной речью необходимо соединить скрытые марковские модели слов в одну общую СММ языка предметной области с учетом вероятностей переходов между словами, которые задаются моделью языка. Каждая модель в последовательности напрямую связана с элементом, лежащим в ее основе. Этими элементами могут быть целые слова или части слов, такие как фонемы. На рисунке 4.8 показана сеть, в которой каждое слово определено как последовательность скрытых

марковских моделей, основанных на фонемах, и все слова замкнуты в петлю (цикл). В этой сети кружками показаны СММ, а прямоугольниками – состояния конца слова.

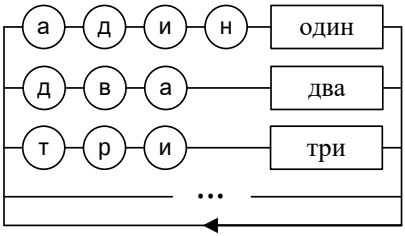


Рис. 4.8. Пример объединенной СММ для метода прохождения маркеров

Таким образом, распознающая сеть в итоге состоит из состояний скрытой марковской модели, соединенных переходами. В ней можно выделить три различных уровня: слов, фонем и состояний модели. На рисунке 4.9 показана эта иерархия.

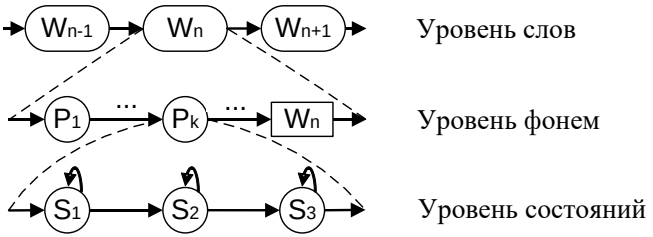


Рис. 4.9. Три уровня описания распознающей сети

Для распознавания слитной речи используется модифицированный алгоритм Витерби, называемый метод передачи маркеров (англ. token passing method) [63]. Метод передачи маркеров определяет прохождение возможных путей по состояниям объединенной СММ. В начало каждого слова ставится маркер и применяется итеративный алгоритм оптимизации Витерби, сдвигая маркер на каждом шаге и вычисляя для него акустическую вероятность. Предположим, в каждом состоянии j скрытой марковской модели в момент времени t находится отдельный маркер, который содержит

значение логарифма вероятности $\psi_j(t)$ пройденной части пути. Этот маркер отображает соотношение между наблюдаемой последовательностью от o_1 до o_t и моделью, позволяющее заключить, что модель находится в состоянии j в момент времени t. Для вычисления вероятности на каждом шаге алгоритма используется рекурсивная формула:

$$\psi_j(t) = \max_i \left\{ \psi_i(t-1) + \log\left(a_{ij}\right) \right\} + \log\left(b_j\left(\mathbf{o}_t\right)\right) \qquad (3)$$

Формула (3) используется в алгоритме, который выполняется в каждый момент времени t для каждого маркера. Ключевые шаги рекурсивного алгоритма следующие:

1) копия каждого маркера, находящегося в состоянии i, должна пройти через следующее состояние j, тогда приращение логарифма вероятности в маркере будет равняться $\log(a_{ij})+\log(b_j(\mathbf{o}_t))$;

2) проверка маркеров в каждом состоянии и удаление всех маркеров, кроме маркеров с самой высокой вероятностью.

При достижении состояния конца некоторого слова в маркер записывается его индекс, а при выходе из каждого состояния маркеры размножаются (копированием) по числу дальнейших переходов в модели. При этом в маркер записывается его путь (история) через сеть. Когда маркер переходит от выходного состояния одного слова к входному состоянию другого, переход представляет собой потенциальную границу слов, которая и записывается в историю маркера. В итоге после обработки всей последовательности векторов наблюдений выбирается маркер, имеющий наибольшую вероятность. Когда наилучший маркер достигает конца обрабатываемого сигнала (последовательности наблюдений), то путь, которым он проходит через сеть, известен в виде истории (хранящейся в маркере), и из маркера считывается последовательность пройденных слов, которая и является гипотезой распознавания фразы. Данная методика распознавания слитной речи эффективно используется в настоящее время для автоматического распознавания речи.

4.4 Применение нейронных сетей для акустического моделирования

При акустическом моделировании искусственные нейронные сети обычно используют совместно со скрытыми марковскими моделями (СММ) [22] при этом СММ обеспечивают возможность моделирования долговременных зависимостей, а ИНС – возможность дискриминантного обучения [83]. Основными методами объединения являются: 1) построение гибридных моделей СММ/ИНС; 2) построение тандемных моделей.

Архитектура гибридной системы представлена на рисунке 4.10.

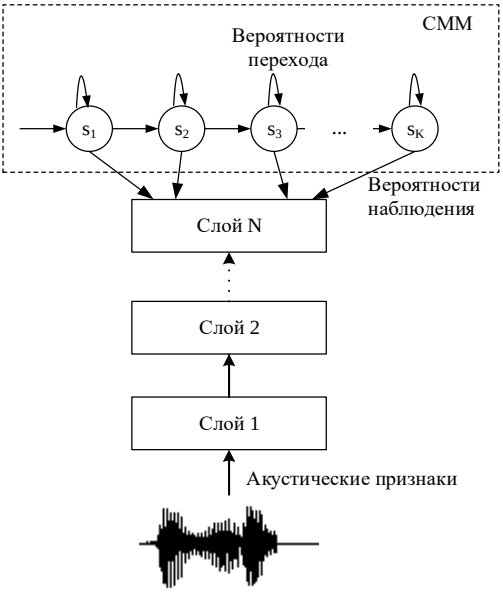


Рис. 4.10. Архитектура гибридной СММ/ИНС модели

ИНС обучается предсказывать апостериорные вероятности каждого контекстно-зависимого состояния при данных акустических наблюдениях. На этапе распознавания речи вместо выходных вероятностей СММ используются так называемые псевдо-вероятности, которые получаются путем деления вероятностей, полученных с выхода ИНС, на априорные вероятности каждого состояния.

В методе тандема выходные данные нейронной сети используются как дополнительный поток признаков для обучения СММ. Архитектура модели, использующей метод тандема, представлена на рисунке 4.11.

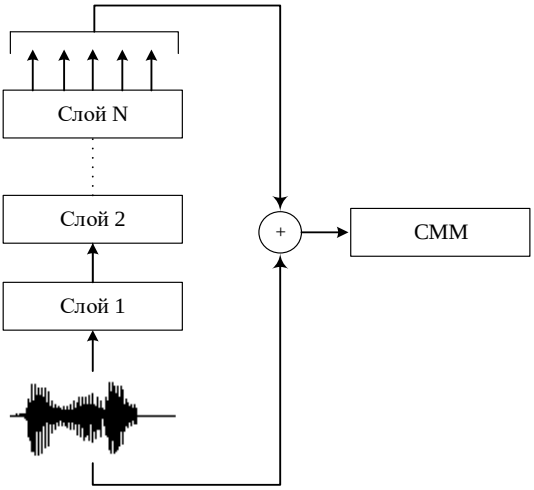


Рис. 4.11. Архитектура модели с использованием тандемного метода

Для акустического моделирования обычно используются глубокие нейронные сети прямого распространения, в нескольких научных работах было показано, что применение в гибридных акустических моделях ИНС с временными задержками позволяет получить большую точность распознавания, чем использование обычной ИНС [30, 41]. Поскольку длительные временные зависимости моделируются с помощью СММ, нет необходимости в применение рекуррентных ИНС, однако может использоваться любая архитектура нейронной сети, описанная в разделе 2, а также их модификации.

4.5 Вопросы по разделу 4

1. Для чего необходимо параметрическое представление речи в системах автоматического распознавания?
2. Что такое дискретизация и квантование речевого сигнала?

3. Какая оконная функция чаще всего используется в ходе параметрического представления речевого сигнала?

4. Как вычисляются мел-частотные кепстральные коэффициенты для сегментов речевого сигнала?

5. Вычислите значение f_{mel} по мел-частотной шкале для частоты f=6,3 кГц.

6. С помощью каких параметров задается гауссовское (нормальное) распределение переменной?

7. Назовите основные задачи, которые решаются с использованием СММ, для распознавания речи.

8. В чем состоит цель обучения акустических моделей?

9. Опишите суть метода передачи маркеров для распознавания слитной речи.

10. Назовите основные методы объединения СММ и ИНС при создании акустической модели.

5 ЯЗЫКОВОЕ МОДЕЛИРОВАНИЕ РЕЧИ

5.1 Статистические модели на основе *n*-грамм

Для задачи распознавания слитной речи с большим словарем необходима модель языка для генерации грамматически правильных и осмысленных гипотез произнесенной фразы. Одной из наиболее эффективных моделей естественного языка является статистическая модель на основе *n*-грамм, цель которой состоит в оценке вероятности появления цепочки слов $\mathbf{W}=(w_1,w_2,\ldots,w_m)$ в некотором тексте. *n*-граммы представляют собой последовательность из *n* элементов (например, слов), а *n*-граммная модель языка используется для предсказания элемента в последовательности, содержащей *n-1* предшественников [70]. Эта модель основана на предположении, что вероятность какой-то определенной *n*-граммы, содержащейся в неизвестном тексте, можно оценить, зная, как часто она встречается в некотором обучающем тексте.

Вероятность $P(w_1,w_2,\ldots,w_m)$ можно представить в виде произведения условных вероятностей входящих в нее *n*-грамм [44]:

$$P\left(w_1, w_2, ..., w_m\right) = \prod_{i=1}^{m} P\left(w_i \mid w_1, w_2, ..., w_{i-1}\right)$$

или аппроксимируя $P(w)$ при ограниченном контексте длиной *n*-1:

$$P\left(w_1, w_2, ..., w_m\right) \cong \prod_{i=1}^{m} P\left(w_i \mid w_{i-n+1}, w_{i-n+2}, ..., w_{i-1}\right)$$

Вероятность появления *n*-граммы вычисляется на практике следующим образом:

$$P\left(w_i \mid w_{i-n+1}, ..., w_{i-1}\right) = \frac{C\left(w_{i-n+1}, ..., w_i\right)}{C\left(w_{i-n+1}, ..., w_{i-1}\right)}$$

где C – количество появлений последовательности в обучающем корпусе.

Тривиальной языковой моделью (ЯМ) является нульграммная ($n=0$) модель, которая предполагает, что каждое слово может следовать за любым другим словом. Тогда вероятность появления слова определяется как [33]:

$$P(w_i) = \frac{1}{|V|},$$

где $|V|$ – размер словаря системы автоматического распознавания речи.

Униграммная ЯМ ($n=1$) определяет вероятность появления i-го слова $P(w_i)$ в тексте. На практике обычно используются биграммная ($n=2$) модель, определяющая вероятность появления пар слов $P(w_i|w_{i-1})$, и триграммная ($n=3$) модель языка, которая определяет вероятность появления троек слов в сказанной фразе $P(w_i|w_{i-2},w_{i-1})$.

Диаграмма создания модели языка показана на рисунке 5.1. Модель языка создается по текстовому корпусу, который должен быть достаточно большого объема и соответствовать предметной области.

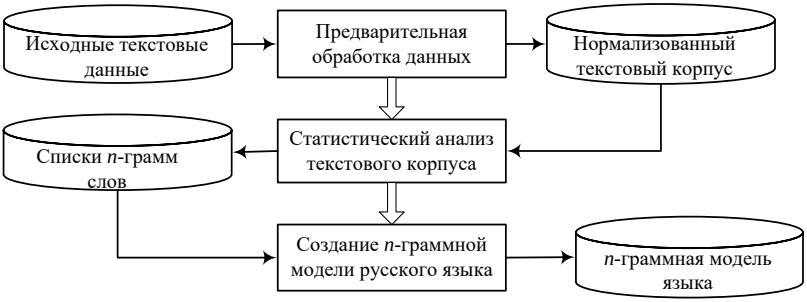


Рис. 5.1. Диаграмма процесса создания ЯМ

Вначале выполняется предварительная обработка текстового материала, которая включает в себя разделение текста на

отдельные предложения, замену заглавных букв на строчные, удаление знаков препинания и т.п. Затем производится статистический анализ текстового корпуса, в результате которого создается список *n*-грамм слов с частотой их появления в обучающем текстовом корпусе. Затем по списку *n*-грамм создается модель языка.

Существует несколько форматов, которые используются для записи ЯМ, самым распространенным является формат ARPA, фрагмент биграммной ЯМ в формате ARPA показан на рисунке 5.2

```
...
\data\
ngram 1=586
ngram 2=930

\1-grams:
-99.0000 <UNK>              0.0000
-99.0000 </s>               0.0000
-1.0263 <s>  -2.5633
-99.0000 автомат            0.0000
-3.0228 автомобилей         -0.7722
-3.1198 автопрома           -0.6990
-2.9437 адекватным          -0.8446
-2.9437 ажиотажа            -0.8446
…

\2-grams:
-1.7933 <s> боже
-1.3161 <s> в
…
-1.0624 в том
-1.5017 в частности
-0.2218 вам ничего
-0.5229 вам предстоит
…
\end\
```

Рис. 5.2. Фрагмент ЯМ в формате ARPA

В самом начале в ЯМ в формате ARPA указывается, сколько в модели языка униграмм и биграмм, затем идет список униграмм, слева от униграммы указывается значение десятичного логарифма вероятности ее появления, справа – коэффициент возврата (англ. back-off weight), который применяется в тех случаях, когда некоторая *n*-грамма отсутствует в обучающем корпусе или частота ее появления очень низкая, тогда вместо нее

используется вероятность (n-1)-граммы, умноженная на коэффициент возврата (процедура возврата описана ниже). Затем идет список биграмм с вероятностями их появления.

Одной из проблем создания статистических моделей языка является неполнота обучающих данных. Каким бы большим ни был обучающий корпус, он не сможет охватить все возможные n-граммы языка. При этом нелогично полностью отказываться от гипотезы, которая обладает очень большой акустической вероятностью только на основании того, что одна из n-грамм в гипотезе не встретилась ни разу в обучающем корпусе. Для решения этой проблемы используются алгоритмы сглаживания (англ. smoothing) и возврата (англ. back-off).

Идея сглаживания заключается в том, что берется часть вероятностной массы у встретившихся в корпусе n-грамм, и распределяется по всем возможным комбинациям слов из словаря, составляющим множество невстреченных n-грамм [95].

Суть методики возврата состоит в том, что, когда некоторая n-грамма отсутствует в обучающем корпусе, тогда вместо нее используется вероятность (n-1)-граммы, умноженная на коэффициент возврата. Например, вычисление вероятности триграммы при использовании процедуры возврата:

$$P\left(w_i \mid w_{i-2}w_{i-1}\right) = \begin{cases} \tilde{P}\left(w_i \mid w_{i-2}w_{i-1}\right), C\left(w_{i-2}w_{i-1}w_i\right) > 0 \\ \alpha\left(w_{n-2}^{n-1}\right) P\left(w_i \mid w_{i-1}\right), \text{в противном случае} \end{cases}$$

Если рассматриваемая триграмма, т.е. последовательность из трех слов, встретилась в обучающем корпусе ($C(w_i|w_{i-2}w_{i-1})>0$), то берется вероятность этой триграммы, посчитанная из обучающего корпуса $\tilde{P}\left(w_i \mid w_{i-2}w_{i-1}\right)$. Если такой триграммы не было, то берется вероятность биграммы, умноженная на коэффициент возврата (α). Коэффициент возврата необходим для корректного распределения остаточной вероятности n-грамм в соответствии с распределением вероятности (n-1)-грамм. Если не вводить α, оценка будет ошибочной, т.к. не будет выполняться условие, что сумма вероятностей равна 1.

5.2 Оценка моделей языка

Для анализа качества созданных моделей языка вычисляют информационную энтропию и коэффициент неопределенности модели языка для тестового текстового корпуса. Информационная энтропия — мера хаотичности информации, неопределенность появления какого-либо символа первичного алфавита [97]. При отсутствии информационных потерь она численно равна количеству информации на символ передаваемого сообщения. Поскольку тексты на естественном языке могут рассматриваться в качестве информационного источника, энтропия вычисляется по следующей формуле [33]:

$$H = -\lim_{m\to\infty}\frac{1}{m}\sum_{w_1,\ldots,w_m}\left(P\left(w_1,\ldots,w_m\right)\log_2 P\left(w_1,\ldots,w_m\right)\right)$$

Это суммирование делается по всем возможным последовательностям слов текстового корпуса. Но поскольку язык является эргодическим источником информации (случайный процесс эргодичен [97], если все его статистические характеристики с вероятностью, сколь угодно близкой к единице, можно предсказать по одной реакции из ансамбля с помощью усреднения по времени) [33], выражение для вычисления энтропии будет выглядеть следующим образом:

$$\hat{H} = -\frac{1}{m}\log_2 P\left(w_1,\ldots,w_m\right)$$

Коэффициент неопределенности является параметром, по которому оценивается сложность *n*-граммных моделей языка, и вычисляется следующим образом [33]:

$$PP = 2^{\hat{H}} = \hat{P}\left(w_1,\ldots,w_m\right)^{-\frac{1}{m}}$$

где $\hat{P}\left(w_1,\ldots,w_m\right)$ — вероятность последовательности слов $(w_1,\ldots,w_m)$.

Коэффициент неопределенности показывает, сколько в среднем различных наиболее вероятных слов может следовать за данным словом. Чем меньше коэффициент неопределенности, тем лучше модель языка, однако сравнивать надо одинаковые по размеру модели, естественно, что коэффициент неопределенности модели с большим размером словаря будет больше, чем у модели с меньшим. Оценка ЯМ должна производиться на текстовых данных, которые не использовались для обучения.

5.3 Разновидности статистических моделей языка

Существует несколько вариантов организации ЯМ, основанных на статистическом анализе текста. Модели, основанные на классах (англ. Class-based models), используют функцию, которая отображает каждое слово w_i на класс c_i: f_i: $w_i \rightarrow f(w_i)=c_i$. В этом случае вначале вычисляется распределения вероятности для классов, затем – распределение вероятности для слов, которые относятся к соответствующему классу. Вероятность появления слова определяется следующим образом:

$$P(w_i|w_{i-n+1}\dots w_{i-1}) = P(w_i|c_i)P(c_i|c_{i-n+1}\dots c_{i-1})$$

где w_i – текущее слово, c_i – класс, к которому принадлежит слово w_i.

Существует несколько методов для кластеризации слов. Наиболее простым способом является разделение слов на классы в соответствии с их частотой появления в обучающем текстовом корпусе [35]. Кластеризация слов может выполняться, основываясь на контексте, в котором эти слова появляются. Подобный метод был описан в работе [8], в которой было предложено в начале отнести каждое слово к отдельному классу и вычислить среднюю взаимную информацию между соседними классами, а затем объединить классы, для которых функция потерь средней взаимной информации будет наименьшей. В работе [36] кластеризация на основе непрерывного мешка слов (англ. Continuous Bag-of-Words) была выполнена посредством создания векторов представления слов и кластеризации их с помощью метода К-средних. Кроме того, для кластеризации слов

могут использоваться методы, основанные на данных. Например, в работе [13] кластеризация была выполнения с помощью набора правил, описывающих явления фонетической редукции и ассимиляции в разговорной речи. В работе [20] кластеризация слов была выполнения для извлечения метаданных документов, при этом классы были сформированы по базам данных различных предметных областей, а также с учетом орфографических признаков слов. Еще одним способом кластеризации слов является использование информации о грамматических признаках слов. Так в работе [49] слова вначале делились на классы в соответствии с их частью речи, а затем классы, размер которых был больше некоторого заранее заданного значения, случайным образом делились на более мелкие.

n-граммые модели обычно имеют ограниченный контекст *n=2,3,4,5,* потому что с увеличением *n* очень быстро растет число параметров модели. Интервальные модели языка (англ. distance models) помогают включить больший контекст, чем *n*-граммы, но величина коэффициента неопределенности модели остается того же порядка, как у *n*-грамм. Например, биграммная интервальная модель может быть задана следующим образом [55]:

$$P\left(w_i \mid w_{i-M+1},...,w_{i-1}\right) = \sum_{m=1}^{M-1} \lambda_m P_m\left(w_i \mid w_{i-m}\right),$$

где M – это предопределенное число моделей, $P_m(w_i|w_{i-m})$ - биграммная модель с пропуском m-1, λ_m – весовые параметры модели при условии $\sum_{m=1}^{M-1} \lambda_m = 1$. Значение весовых коэффициентов λ_m определяется как зависимость от расстояния между словами w_i и w_{i-m} (с увеличением расстояния величина весового коэффициента уменьшается).

Триггерные модели (англ. Trigger models) — другой тип моделей, которые моделируют взаимоотношение пар слов в более длинном контексте. В этом методе появление инициирующего слова в истории увеличивает вероятность другого слова, называемого целевым, с которым оно связано.

Вероятность пар слов может быть определена следующим образом [55]:

$$P_{a\to b}\left(b \mid a \in h\right) = \frac{C\left(a \in h, b\right)}{C\left(a \in h\right)},$$

здесь a – это инициирующее слово, b – целевое слово, функция C определяет подсчет события в текстовом корпусе, h – история некоторого ограниченного размера для слова b, то есть слова, предшествующие в тексте слову b. Полная триггерная модель может быть определена следующим образом [55]:

$$P\left(w_i \mid w_M, ..., w_{i-1}\right) = \frac{1}{M}\sum_{m=1}^{M}\alpha\left(w_i, w_{i-m}\right),$$

$$\alpha\left(b, a\right) = \frac{P_{a\to b}\left(b \mid a \in h\right)}{\sum_{w} P_{a\to w}\left(w \mid \alpha \in h\right)},$$

здесь M определяет длину цепочки слов в анализируемой истории h.

Упрощенной версией триггерных пар является кэш-модель (англ. cache model) [66]. Кэш-модель увеличивает вероятность появления слова в соответствии с тем, как часто данное слово употреблялось в истории, поскольку считается, что, употребив конкретное слово, диктор будет использовать это слово еще раз либо из-за того, что оно является характерным для конкретной темы, либо потому что диктор имеет тенденцию использовать это слово в своем лексиконе. Кэш-модель можно рассматривать как простую n-граммную модель с вероятностями, вычисленными по предшествующей истории слов. Обычная униграммная кэш-модель может определяться как [66]:

$$P_C\left(w_i \mid h\right) = \frac{C\left(w_i, h\right)}{C\left(h\right)} = \frac{\sum_{j=i-D}^{i-1} I\left(w_i = w_j\right)}{\sum_{j=i-D}^{i-1} I\left(w_j \in V\right)},$$

где D – это размер истории h, I – индикаторная функция, V – словарь модели языка. Более развитые кэш-модели объединяются с убывающей функцией, благодаря чему вероятность повторения слова уменьшается с увеличением расстояния от последнего появления слова в тексте:

$$P_{DC}\left(w_i \mid h\right) = \frac{\sum_{j=i-D}^{i-1}\left[I\left(w_i = w_j\right)\cdot d\left(i-j\right)\right]}{\sum_{j=i-D}^{i-1} d\left(i-j\right)},$$

где d – некоторая убывающая функция.

Другим типом ЯМ является модель на основе набора тем (англ. topic mixture models). Текстовый корпус вручную или автоматически делится на предопределенное число тем, и языковые модели создаются отдельно для каждой темы. Полная модель может определяться как [66]:

$$P_{TM}\left(w_i \mid h_i\right) = \sum_{j=1}^{J} \lambda_j \cdot P_j\left(w_i \mid h_i\right),$$

где J – это число тем, и P_J – модель темы j с весом модели λ_j. Веса модели могут быть статическими или динамическими. Если используются динамические веса модели, тогда они устанавливаются для каждого слова w_i в зависимости от предшествующей истории. Такая модель может называться адаптивной моделью.

Модели, основанные на частях слов (англ. subword-based models), используются для языков с богатой морфологией, например, флективных языков [48]. В этом случае слово w разделяется на некоторое число $L(w)$ частей (морфем) с помощью функции $U{:}w{\rightarrow}U(w){=}u^1{,}u^2{,}\ldots{,}u^{L(w)}{,}u^i{\in}\psi$, где ψ – это набор частей слова. Разделение слов на морфемы можно производить двумя путями: при помощи словарных и алгоритмических методов [48]. Преимуществом алгоритмических методов является то, что они опираются лишь

на анализ текста и не используют никаких дополнительных знаний, что позволяет анализировать текст на любом языке. Преимуществом словарных методов является то, что они позволяют получить правильное разбиение слов на морфемы, а не на псевдоморфемные единицы (как в алгоритмических методах), что может быть использовано далее на уровне постобработки гипотез распознавания фраз. В качестве значимых частей слов часто при распознавании речи с большим словарем используют морфемы. Такие модели разработаны для целого ряда синтетических языков, например, финского [23], турецкого [9], венгерского [52], словенского [46], чешского [*39*], русского [74] и т.д. Обычно в русском языке выделяют 6 типов морфем: префикс, корень, интерфикс, суффикс, окончание, постфикс. Пример декомпозиции нескольких слов на морфемы показан на рисунке 5.3.

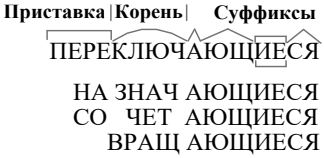


Рис. 5.3. Пример декомпозиции слов на морфемы

Хотя, по определению, *n*-граммные модели языка хранят только *n* слов, существуют модели, которые не ограничивают последовательности слов до определенного *n*, а вместо этого хранят различные последовательности разной длины. Такие модели называют *n*-граммами переменной длины (англ. varigrams) [38]. По существу, они могут рассматриваться как *n*-граммные модели с большим *n* и такими принципами сокращения длины моделей, которые сохраняют только небольшой поднабор всех длинных последовательностей, встретившихся в обучающем тексте.

Также применяется дальнодействующая триграммная модель [88], представляющая собой обычную триграммную модель, в которой разрешены связи между словами, находящимися не только в пределах двух предыдущих слов, но и на большем расстоянии от предсказываемого слова. Лежащая в основе «грамматика» представляет собой множество пар слов,

которые могут быть связаны вместе через несколько разделяющих слов.

В системах распознавания русской речи, для которой характерен нежесткий порядок слов, может использоваться синтаксическо-статистическая ЯМ, позволяющая учесть дальнодействующие связи между словами [27]. Для создания такой модели вначале выполняется статистический анализ текстового корпуса и создается список *n*-грамм слов. Затем производится синтаксический анализ, в ходе которого выявляются грамматически связанные пары слов (синтаксические группы), которые были разделены в тексте другими словами. Такие синтаксические группы добавляются к списку *n*-грамм слов, полученных в ходе статистического анализа текстового корпуса. Диаграмма создания синтаксическо-статистической модели языка показана на рисунке 5.4.

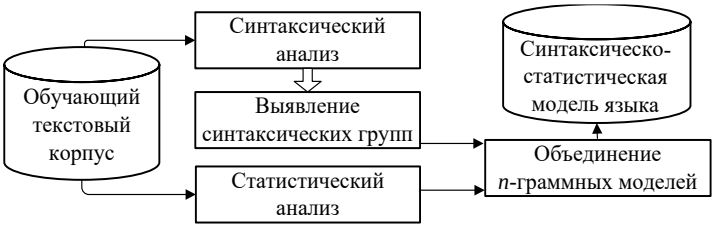


Рис. 5.4. Процесс создания синтаксическо-статистической ЯМ

Для моделирования языков с богатой морфологией может использоваться факторная ЯМ, которая впервые была предложена в работе [6] для моделирования арабского языка и может быть использована для других языков с богатой морфологией. Эта модель объединяет различные признаки слова, называемые факторами, при этом слово представляется как вектор k факторов:

$$w_i = \left(f_i^1, f_i^2, ..., f_i^k \right)$$

В качестве факторов могут использоваться: слово, морфологический класс, основа, корень слова и другие грамматические признаки. Например, словоформа «*схеме*» может быть представлена в виде набора факторов следующим образом: «*W-схеме L-схема S-схем P-сущ M-bc*», где *W* –

словоформа, *L* – лемма, *S* – основа, *P* – часть речи, *M* – метка морфологических признаков, содержащая всю грамматическую информацию о слове (в данном примере она означает, что словоформа является существительным женского рода в единственном числе и дательном падеже).

При создании факторной ЯМ необходимо выбрать подходящий набор факторов, а также граф возврата. В *n*-граммных моделях языка возврат осуществляется путем отбрасывания сначала наиболее дальнего слова, затем второго по дальности слова и т.д. Этот процесс показан на рисунке 5.5а. Для факторной модели языка нет очевидного пути возврата [6], любой фактор может быть опущен на каждом шаге выполнения процедуры возврата, и не является очевидным, какой фактор должен быть опущен первым. Таким образом, возможны несколько путей возврата, в результате получается граф возврата. Пример графа возврата факторной модели языка представлен на рис. 5.5б. Граф показывает все возможные пути возврата, при которых один фактор опускается на каждом шаге процедуры возврата.

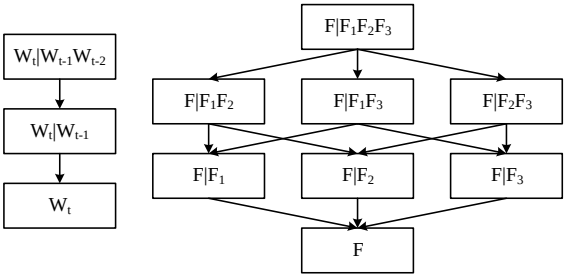


Рис. 5.5. Пути возврата для n-граммной и факторной ЯМ: а) путь возврата для 3-граммной ЯМ; б) граф возврата для трехфакторной ЯМ

Следует отметить, что в одной системе распознавания речи можно использовать сразу несколько моделей языка, для этого используется метод линейной интерполяции, при этом объединяются не сами ЯМ, а полученные с их помощью вероятности. Итак, под интерполяцией моделей языка понимается линейная комбинация вероятностей слов,

полученных от разных моделей, с учетом весовых коэффициентов каждой модели. Таким образом, итоговая вероятность цепочки слов из i слов будет определяться следующим образом:

$$P_{linear}\left(w_i \mid w_1,...,w_{i-1}\right) = \sum_{m=1}^{M} \lambda_m P_{M_m}\left(w_i \mid w_1,...,w_{i-1}\right),$$

где M – количество используемых моделей языка, λ_m – весовой коэффициент каждой модели.

Таким образом, вначале можно получить вероятности для некоторой фразы с использованием нескольких ЯМ, а затем сложить полученные результаты, предварительно умножив каждый из них на весовой коэффициент соответствующей модели. Весовые коэффициенты чаще всего выбираются эмпирически. Сумма весовых коэффициентов должна быть равна единице, чтобы обеспечить то, чтобы сумма всех вероятностей равнялась единице:

$$\sum_{m=1}^{M} \lambda_m = 1$$

Статистические n-граммные ЯМ, а также их различные модификации показывают хорошие результаты при их использовании в системах автоматического распознавания речи, однако в последнее время все большую популярность приобретают модели языка на основе ИНС. Нейросетевые модели языка будут рассмотрены в следующем разделе.

5.4 Применение нейронных сетей для языкового моделирования

Для языкового моделирования в системах автоматического распознавания речи могут использоваться ИНС как прямого распространения, так и рекуррентные. Архитектура ЯМ на базе ИНС прямого распространения представлена на рисунке 5.6 [15]. В ИНС прямого распространения входной слой сети представляет собой историю из n-1 слов, предшествующих данному слову. Каждое слово из словаря ассоциировано с

вектором длиной, равной размеру словаря, где только одно значение, соответствующее индексу данного слова в словаре, равно 1, а все остальные значения равны 0. Слой, сформированный путем объединения векторов слов, называется проекционным слоем.

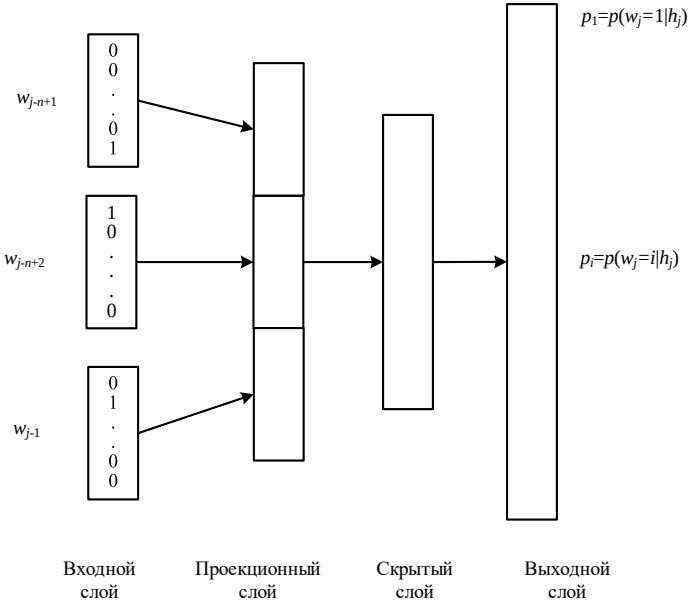


Рис. 5.6. Архитектура ЯМ на базе ИНС прямого распространения

Основным недостатком таких сетей является то, что для предсказания слова они используют предшествующий контекст определенной длины. Поэтому для языкового моделирования более предпочтительным является использование рекуррентных ИНС. Архитектура ЯМ на основе рекуррентной ИНС с одним срытым слоем показана на рисунке 5.7.

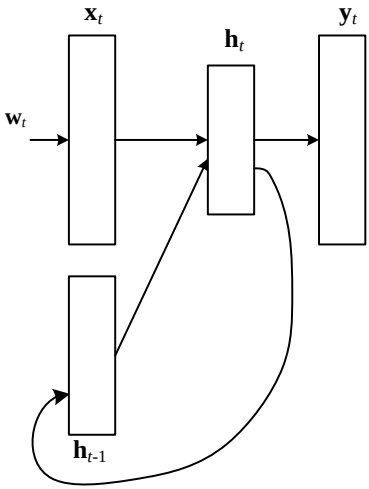


Рис. 5.7. Архитектура ЯМ на базе рекуррентной ИНС

Входной слой ИНС состоит из вектора $\mathbf{x}_t$, который является объединением вектора $\mathbf{w}_t$, представляющим собой текущее слово, и вектора $\mathbf{h}_{t-1}$, который представляет собой выходные значения скрытого слоя, полученные на предыдущем шаге. Размер $\mathbf{w}_t$ равен размеру словаря. Выходной слой $\mathbf{y}_t$ имеет такую же размерность, как и $\mathbf{w}_t$, и после обучения нейронной сети представляет собой вероятностное распределение следующего слова при данном предыдущем слове и состоянии скрытого слоя в предыдущий временной шаг. Размер скрытого слоя выбирается эмпирически. В рекуррентной искусственной нейронной сети скрытый слой хранит всю предыдущую историю, таким образом, размер контекста неограничен.

Также для языкового моделирования могут использоваться одно- и двунаправленные сети LSTM. Архитектура ЯМ на базе LSTM представлена на рисунке 5.8, где $\mathbf{w}_t$ - входное слово в момент времени t; $\mathbf{h}_t$ – состояние скрытого слоя, $\mathbf{c}_t$ – состояние ячейки LSTM.

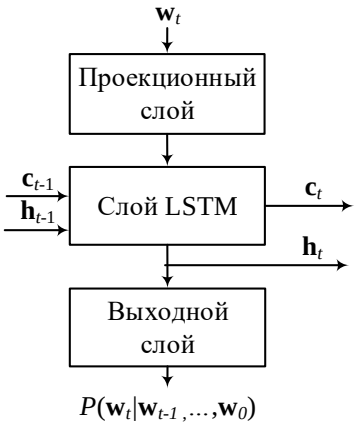


Рис. 5.8. Архитектура модели языка, основанной на LSTM

Модели языка на базе ИНС имеют существенно меньший коэффициент неопределенности по сравнению с триграммной моделью, однако их обучения занимает длительное время. Одним из способов сократить время обучения является использовать модель языка, основанную на классах.

Модель языка на базе ИНС проблематично подключать на этапе декодирования речевого сигнала, поэтому обычно декодирование выполняют с применением *n*-граммной ЯМ и при этом в качестве результата распознавания выводят не одну наилучшие гипотезу, а список лучших гипотез (N-best list). Список лучших гипотез распознавания представляет собой список гипотез с наибольшими вероятностями, посчитанными с применением акустической и языковой моделей. Пример списка 10 лучших гипотез распознавания для фразы «*Находку в наших архивах можно назвать подарком судьбы*» представлен на рисунке 5.9.

```
#1 -16459.691406 -562.112976 <s> находку наших архивах можно назвать подарком судьбы </s>
#2 -16476.886719 -548.556274 <s> находку в наших архивах можно назвать подарком судьбы </s>
#3 -16497.082031 -554.726868 <s> находку наших архивов можно назвать подарком судьбы </s>
#4 -16485.552734 -604.037048 <s> в находку наших архивах можно назвать подарком судьбы </s>
#5 -16502.751953 -597.559387 <s> в находку в наших архивах можно назвать подарком судьбы </s>
#6 -16516.050781 -590.159668 <s> и находку в наших архивах можно назвать подарком судьбы </s>
#7 -16503.970703 -603.337463 <s> а находку наших архивах можно назвать подарком судьбы </s>
#8 -16521.166016 -589.780762 <s> а находку в наших архивах можно назвать подарком судьбы </s>
#9 -16498.855469 -615.080627 <s> и находку наших архивах можно назвать подарком судьбы </s>
#10 -16522.945313 -596.650940 <s> в находку наших архивов можно назвать подарком судьбы </s>
```

Рис. 5.9. Пример списка из 10 лучших гипотез распознавания

На рисунке 5.9 первая цифра слева означает порядковый номер гипотез, затем идет значение десятичного логарифма акустической вероятности гипотезы, а затем значение десятичного логарифма вероятности по модели языка, далее идет гипотеза распознавания произнесенной диктором фразы. В реальности обычно используется список из 100-500 гипотез. Для каждой из гипотез определяется вероятность с использованием нейросетевой модели языка, также может быть выполнена интерполяция нейросетевой и *n*-граммной моделей языка. Далее старое значение вероятности по ЯМ заменяется новым значением, полученным с помощью нейросетей модели, и после этого происходит переранжирование гипотез и определение гипотезы с наибольшей вероятностью. Процесс декодирования речевого сигнала с использованием нейросетевой ЯМ показан на рисунке 5.10.

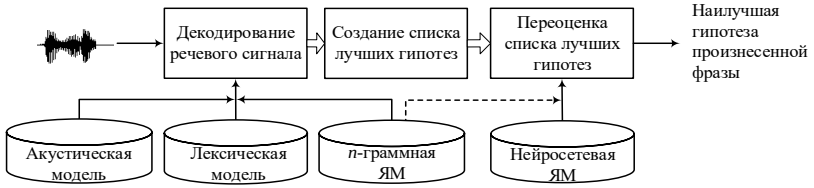


Рис. 5.10. Процесс декодирования речевого сигнала с использованием нейросетевой ЯМ

Кроме того, нейросетевые ЯМ могут использоваться для генерации текста с целью аугментации текстовых данных для обучения *n*-граммной ЯМ. Подобный подход описан в работе [50].

Следует отметить, что все рассмотренные в данном разделе варианты статистических и нейросетевых языковых моделей обладают универсальностью и применимы к большинству естественных языков.

5.5 Вопросы по разделу 5

1. Для чего системам автоматического распознавания речи необходимы модели языка?
2. Что такое *n*-граммы?

3. В чем состоит суть *n*-граммного статистического моделирования языка?

4. Как создается *n*-граммная модель языка с использованием обучающих текстовых баз данных?

5. В чем состоит отличие униграммной модели языка от нульграммной?

6. Посчитайте вероятность появления слова в нульграммной модели языка для словаря произвольного размера от 25 от 250 слов.

7. Какие разновидности статистических моделей языка вы знаете?

8. Какие виды статистических моделей языка позволяют моделировать дальнодействующие связи между словами?

9. В чем состоит суть триггерной *n*-граммной модели языка?

10. Для каких языков применяют морфемные модели?

11. Что такое коэффициент неопределенности модели языка? Приведите формулу вычисления коэффициента неопределенности.

12. Как выполняется интерполяция моделей языка?

13. В чем состоит преимущество ЯМ на базе рекуррентных ИНС?

14. Что такое список *N* лучших гипотез распознавания (*N*-best list)?

6 ИНТЕГРАЛЬНЫЕ МОДЕЛИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ

6.1 Основные особенности интегральных моделей распознавания речи

Стандартные модели распознавания речи состоят из нескольких компонентов (акустическая, лексическая и языковая модели), которые обучаются независимо, и ошибки в одних компонентах могут вызывать ошибки в других. Сценарий стандартной системы состоит из множества шагов, что требует большого объема памяти для хранения, например, обученных языковых моделей, и не позволяет использовать системы локально на различных устройствах, а требует удаленных вычислений на серверах.

Благодаря развитию искусственных нейронных сетей появился новый подход к построению систем распознавания речи, при котором обучение выполняется так, что только одна модель генерирует необходимые выходные данные без использования других компонентов. Такие модели называются интегральными (англ. end-to-end). Обычно в качестве интегральных моделей используются глубокие ИНС. Таким образом, интегральные системы распознавания речи преобразуют речевой сигнал в последовательность букв (символов), составляющих слова, используя при этом одну глубокую ИНС, тем самым сокращается скорость обработки и объем требуемой памяти по сравнению со стандартными системами распознавания речи, состоящими из отдельных компонентов [85]. На рисунке 6.1 изображена схема работы интегральной системы.

Рис. 6.1. Интегральная система распознавания речи

Недостатком интегральных моделей является потребность в большом количестве размеченных речевых данных для обучения (существенно большем, чем для стандартных систем).

Основными моделями для разработки интегральных систем распознавания речи являются модель на основе коннекционной

временной классификации (англ. Connectionist Temporal Classification; CTC) и модель кодер-декодер (англ. encoder-decoder) с механизмом внимания (англ. attention mechanism). Данные модели предназначены для задач, где длины входной и выходной последовательностей являются переменными. В следующих разделах представлено описание данных моделей.

6.2 Модель на основе коннекционной временной классификации

Модель с использованием коннекционной временной классификации работает на уровне распознавания отдельных букв. Данный тип моделей преобразует звуковой сигнал в последовательность букв, а затем удаляет из нее ненужные буквы, пробелы и повторы. Выходной слой нейронной сети содержит по одному блоку для каждого символа выходной последовательности (букв, фонем, знаков препинания) и еще один для дополнительного символа «пропуск» («blank»), соответствующего пустому выходному символу. Выходной вектор $\mathbf{w}_m$ нормализуется с помощью функции softmax [7], которая интерпретируется как вероятность появления символа (или «пропуска») с индексом k в момент времени m:

$$P(k,m\,|\,\mathbf{x}) = \frac{e^{w_m^k}}{\sum_{k'=0}^{|w_m|} e^{w_m^{k'}}}$$

где $\mathbf{x}$ – входная последовательность признаков длинной T, $w_m{}^k$ – k-ый элемент вектора $\mathbf{w}_m$. Активационная функция softmax обеспечивает сумму вероятностей элементов выходного вектора $\mathbf{w}_m$, равную единице.

Обозначим α — последовательность из индексов «пропусков» и символов длины T для выравнивания. Вероятность $P(\alpha|\mathbf{x})$ можно представить, как произведение вероятностей появления символов в каждый момент времени:

$$P(\alpha\,|\,\mathbf{x}) = \prod_{t}^{T} P(\alpha_t, t\,|\,\mathbf{x})$$

Для данной выходной последовательности $|\mathbf{w}_m|$ существует столько возможных выравниваний, сколько способов расставить «пропуски» между символами.

Обозначим символом «—» – «пропуск». Например, выравнивания (а,—,б,в,—,—) и (—, —, а, —, б, в) соответствуют последовательности (а,б,в). Когда одинаковые символы появляются последовательно, то эти повторы удаляются: (а,б,б,б,в,в) и (а,—,б,—,в,в) соответствуют (а,б,в). Обозначим В – оператор, который удаляет сначала все повторы, а затем – «пропуски». Полная вероятность выходной последовательности **w** равна сумме вероятностей всех возможных соответствующих выравниваний:

$$P(\mathbf{w} \mid \mathbf{x}) = \sum_{\alpha \in B^{-1}(w)} P(\alpha \mid \mathbf{x})$$

где B^{-1} – оператор обратный к B.

Эта сумма по всем возможным выравниваниям позволяет нейронной сети обучаться на несегментированных данных. То есть, не зная точное расположение меток, можно выполнить суммирование по всем расположениям, где они могут быть. Эта сумма может быть вычислена с помощью динамического программирования [17].

Пусть $\mathbf{w}^*$ — целевая последовательность слов, тогда нейронная сеть может быть обучена минимизировать CTC функцию:

$$\text{CTC}(x) = -\log P(\mathbf{w}^x|\mathbf{x})$$

Нейронная сеть может быть обучена с помощью любого оптимизационного алгоритма, использующего градиент. На рисунке 6.2 представлена схема CTC модели. Кодером может быть любая нейронная сеть.

Рис. 6.2. CTC система распознавания речи

В [17] предложен CTC алгоритм прямого-обратного хода, который использует алгоритм динамического программирования, похожий на алгоритм прямого обратного хода для CMM [44]. Основная идея этого алгоритма состоит в том, что сумма по всем выравниваниям разбивается на сумму по выравниваниям соответствующих префиксам их выходных последовательностей. Эта сумма может быть эффективно вычислена с помощью рекурсивных прямых и обратных переменных.

Также в [17] была предложена метрика количества неверных меток (англ. label error rate; LER) для временного классификатора h как среднее нормализованное расстояние Левенштейна между выходом классификатора и истинным результатом:

$$LER(h,S') = \frac{1}{|S'|}\sum_{(x,\mathbf{w})\in S'}\frac{dist\left(h(\mathbf{x}),\mathbf{w}\right)}{|\mathbf{w}|},$$

где $dist(h(\mathbf{x}),\mathbf{w})$ – расстояние Левенштейна между последовательностями $h(\mathbf{x})$ и $\mathbf{w}$, S' – тестовая выборка состоящая из пар векторов $(\mathbf{x},\mathbf{w})$. Эту метрику нейронная сеть и пытается минимизировать.

В [17] было представлено два варианта декодирования интегральных CTC-моделей. Первый метод (нахождения наилучшего выравнивания выходной последовательности) основывается на предположении, что наиболее вероятное выравнивание соответствует наиболее вероятной выходной последовательности:

$$\arg\max_{w} P(\mathbf{w} \mid \mathbf{x}) \approx B\left(\alpha^{*}\right),$$

где $\alpha^{*} = \arg\max_{\alpha} P(\alpha \mid x)$. Вычисление наилучшего выравнивания является простой задачей, так как α^{*} — конкатенация наиболее «активных» выходов на каждом временном шаге. Однако это не гарантирует нахождение наиболее вероятной последовательности слов.

Второй метод (метод нахождения префиксов) основывается на факте, что, модифицировав алгоритм прямого-обратного хода, описанный выше, можно эффективно вычислять вероятности последовательных расширений префиксов выходных последовательностей.

В [18] был предложен метод декодирования, использующий алгоритм лучевого поиска (англ. beam search algorithm), который также позволяет интегрировать языковую модель. Предложенный алгоритм похож на алгоритм декодирования для гибридных СММ/ИНС систем, но отличается интерпретацией выхода нейронной сети. В гибридных системах выходные значения нейронной сети интерпретируются как апостериорные вероятности состояний, которые затем комбинируются с вероятностями перехода и СММ. В CTC сети выходные значения нейронной сети сами представляют собой вероятности перехода.

CTC модели не лишены недостатков. Во многих работах было отмечено, что при отсутствии языковых моделей CTC-модели часто ошибаются в символах распознанных последовательностей, хотя звучание сохраняется правильным. Также CTC-модели все еще используют предположение о независимости наблюдаемых переменных. Это значит, что CTC-сети требуется языковая модель, при добавлении которой, ошибка распознавания значительно уменьшается [34].

6.3 Модель на основе архитектуры кодер-декодер с механизмом внимания

Модель кодер-декодер представляет собой две нейронные сети. Кодер (англ. encoder) – это нейронная сеть, которая преобразует входную последовательность $\mathbf{X}=(x_1,\ldots,x_N)$ в некоторую промежуточную последовательность $\mathbf{H}=(h_1,\ldots,h_N)$.

Декодер (англ. decoder) — это обычно рекуррентная ИНС, которая использует эту промежуточную последовательность для генерации выходных последовательностей $\mathbf{Y}=(y_1,\ldots,y_I)$. Вероятность $P(\mathbf{Y}|\mathbf{X})$ определяется по следующей формуле:

$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{I} P(y_i \mid y_{1:i-1}, \mathbf{X})$$

где $y_{1:i\text{-}1}$ – часть выходной последовательности.

В декодере может быть применен механизм внимания, который выбирает часть входной последовательности, которая затем используется для предсказания следующего выходного значения. На рисунке 6.3 изображена схема кодер-декодер модели с механизмом внимания.

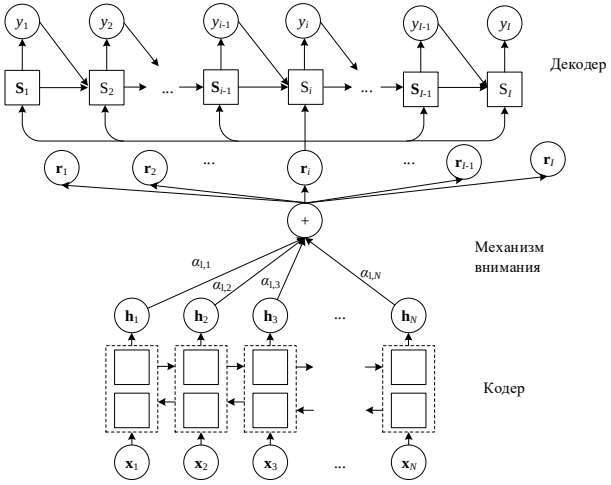


Рис. 6.3. Архитектура модели кодер-декодер с механизмом внимания

Модель кодер-декодер с механизмом внимания работает следующим образом. Кодер преобразует входную последовательность **X** в некоторое промежуточное состояние $\mathbf{h}_n$:

$$\mathbf{h}_n=\text{Encoder}(\mathbf{X})$$

Декодер предсказывает последующий символ y_i в зависимости от предыдущего символа y_{i-1}, вектора скрытого состояния декодера на предыдущем шаге $\mathbf{s}_{i-1}$ и вектора $\mathbf{r}_i$, определяемого следующим образом:

$$\mathbf{r}_i = \sum_{n=1}^{N} \boldsymbol{\alpha}_{in} \left(\mathbf{h}_n \right)$$

где $\boldsymbol{\alpha}_i$ – вектор весов внимания (англ. attention weights), который представляет собой последовательность весов $(\alpha_{i0},\alpha_{i1},\ldots,\alpha_{iN})$. Таким образом, последовательность символов на выходе декодера определяется следующим образом:

$$P(y_i|y_{1:i-1},\mathbf{X})=\text{Decoder}(\mathbf{r}_i, \mathbf{s}_{i-1},y_{i-1})$$

Механизм внимания может быть представлен как метод выравнивания соответствующих фреймов входной последовательности для предсказывания выходной последовательности на конкретном временном шаге. Другими словами, механизм внимания помогает решить, на каких временных фреймах и насколько сильно нужно «сосредоточить внимание» для прогнозирования выходной последовательности на соответствующем временном шаге [26].

Существует несколько вариантов реализации механизма внимания, основными из которых являются следующие [21]:

- внимание на основе скалярного произведения (англ. dot-product attention);
- аддитивное внимание (англ. additive attention);
- внимание по расположению (англ. location-based attention).

Внимание на основе скалярного произведения и аддитивное внимание относятся к одному классу — механизмы внимания по содержанию, в которых внимание вычисляется по значению (содержанию) вектора предыдущего скрытого состояния декодера ($\mathbf{s}_{i-1}$) и вектора, получаемого на выходе кодера ($\mathbf{h}_n$), вне зависимости от их позиции в последовательности. Внимание на основе скалярного произведения вычисляется следующим образом [32]:

$$e_{in} = \mathbf{s}_{i-1}^{N} \mathbf{W}_{\alpha} \mathbf{h}_n$$

$$\boldsymbol{\alpha}_i = \text{softmax}(\mathbf{e}_i)$$

где e_{in} – оценка соответствия i-го скрытого состояния декодера и n-го скрытого состояния кодера, $\mathbf{W}\alpha$ представляет собой матрицу параметров, определяемых в ходе обучения.

В аддитивном механизме внимания оценка $\mathbf{e}_{in}$ вычисляется следующим образом [2]:

$$e_{in} = \mathbf{g}^{\mathrm{T}} \tanh(\mathbf{W}_q \mathbf{s}_{i-1} + \mathbf{W}_{\mathrm{h}} \mathbf{h}_n + \mathbf{b})$$

где $\mathbf{W}_q$ и $\mathbf{W}_h$ – весовые матрицы, $\mathbf{g}$ и $\mathbf{b}$ – векторы параметров.

Механизм внимания по расположению, предложенный в работе [11], учитывает выравнивание, выполненное на предыдущем шаге. Вначале извлекаются векторы $\mathbf{f}_{in}$ для каждой позиции n предыдущего выравнивания $\boldsymbol{\alpha}_{i-1}$ путем выполнения операции свертки с матрицей $\mathbf{F}$:

$$\mathbf{f}_i = \mathbf{F}\boldsymbol{\alpha}_{i-1}$$

Полученные таким образом вектора используются для вычисления e_{in}:

$$e_{in} = \mathbf{g}^{\mathrm{T}} \tanh(\mathbf{W}_q s_{i-1} + \mathbf{W}_{\mathrm{h}} \mathbf{h}_n + \mathbf{W}_f f_{in} + \mathbf{b})$$

Кроме того, может использоваться механизм мультивнимания (англ. multi-head attention), предложенный в работе [56], при котором параллельно вычисляются несколько векторов внимания. Итоговый вектор внимания $\boldsymbol{\alpha}_i^{(MH)}$ вычисляется путем выполнения операции конкатенации всех векторов внимания:

$$\boldsymbol{\alpha}_i^{(MH)} = \text{Concat}(\boldsymbol{\alpha}_{i_1}, ..., \boldsymbol{\alpha}_{i_j}, ...\boldsymbol{\alpha}_{i_M})\mathbf{W}_o$$

где $\mathbf{W}_o$ – матрица весов, M – число векторов внимания. Любой из перечисленных выше типов механизмов внимания может быть реализован как мультивнимание.

В работе [29] была предложена модель, объединяющая CTC модель и кодер-декодер модель с механизмом внимания. В такой модели используются сразу две функции потерь: коннекционной временной классификации и перекрестной энтропии, – значения

которых объединяются с помощью взвешенной суммы, где веса задаются заранее перед обучением или декодированием. Общая функция потерь определяется следующим образом:

$$L = \lambda L_{CTC} + (1 - \lambda) L_{att}$$

где L_{CTC} – функция потерь модели CTC; L_{att} – функция потерь кодер-декодер модели с механизмом внимания; λ – весовой коэффициент CTC модели.

6.4 Модель на основе архитектуры трансформер

Еще одним типом архитектуры нейронных сетей, достаточно успешно применяемым для интегрального распознавания речи, является архитектура трансформер (англ. Transformer) [56]. Данный тип нейронных сетей использует механизм самовнимания (англ. self-attention) и не использует рекуррентные нейронные сети.

Трансформер является развитием архитектуры кодер-декодер с механизмом внимания. Архитектура трансформер показана на рисунке 6.4. Кодер содержит несколько идентичных слоев (в базовой архитектуре - 6), каждый из которых состоит из двух подслоев. Первый подслой представляет собой механизм самовнимания, при этом используется мультивнимание. За слоем мультивнимания располагается ИНС прямого распространения. К каждому подслою добавляются остаточные связи, после которых выполняется нормализация. Выходные данные из последнего слоя кодера поступает на вход всем слоям декодера. Декодер также состоит из нескольких одинаковых слоев, аналогичных слоям кодера (в базовой архитектуре их также 6). Кроме того, декодер также содержит дополнительное маскированное мультивнимание, что помогает декодеру «фокусировать внимание» на соответствующих частях входной последовательности. Блоки кодера могут обрабатывать входную последовательность параллельно, однако декодер работает в авторегрессионной манере.

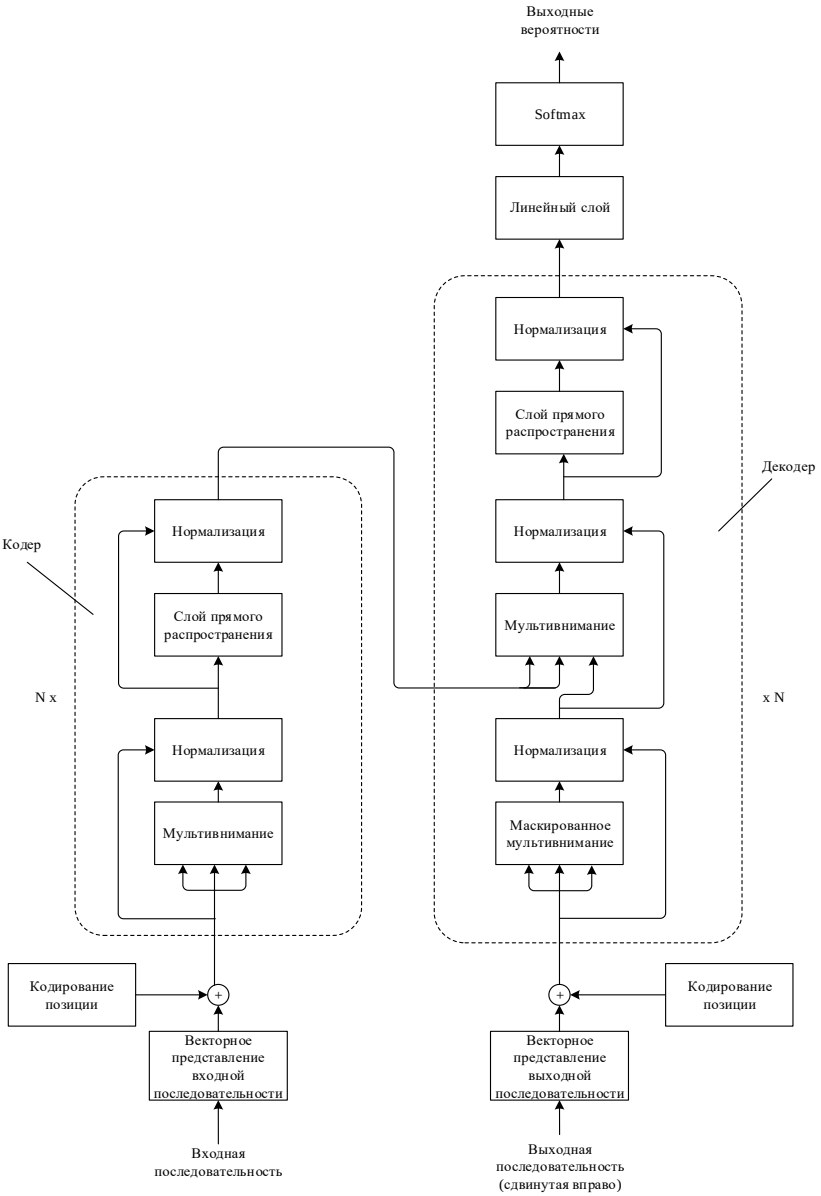


Рис. 6.4. ИНС с архитектурой трансформер, где N – число слоев

Механизм самовнимания определяется следующим образом. Входные данные преобразуется в три вектора: запросов (англ. query), ключей (англ. key) и значений (англ. value). Вычисление этих векторов осуществляется следующим образом:

$$\mathbf{Q}=\mathbf{W}_Q\mathbf{X},$$
$$\mathbf{K}=\mathbf{W}_K\mathbf{X},$$
$$\mathbf{V}=\mathbf{W}_V\mathbf{X},$$

где $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ – вектора запросов, ключей и значений соответственно, $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ – весовые матрицы, которые определяются в ходе обучения. Самовнимание вычисляется следующим образом:

$$\text{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\text{T}}}{\sqrt{d_k}}\right)\mathbf{V},$$

где d_k – размерность вектора ключей.

Одной из проблем создания интегральных систем распознавания речи является необходимость существенно большего объема обучающих речевых данных, чем для стандартного подхода. В следующем разделе будут рассмотрены основные методы обучения интегральные моделей распознавания речи при недостатке обучающих данных.

6.5 Основные методы улучшения работы интегральных систем распознавания речи при недостатке обучающих речевых данных

Основными методами улучшения работы интегральных систем распознавания речи при недостатке обучающих речевых данных являются:

– аугментация речевых данных;

– применение метода переноса знаний (англ. Transfer learning);

– использование внешней модели языка.

Под аугментацией данных понимается увеличение выборки данных для обучения через модификацию существующих

данных. Основными методами аугментации речевых данных являются:

– изменение темпа речи;

– изменение высоты голоса;

– наложение белого шума (белый шум — стационарный шум, спектральные составляющие которого равномерно распределены по всему диапазону задействованных частот);

– синтез речи.

Метод переноса знаний — это метод обучения ИНС, когда знания нейросети, которая была обучена на одной задаче, переносятся на другую задачу. Этот метод используется в том случае, если обучающих данных для целевой задачи мало, но имеется большой обучающий корпус для другой смежной задачи. В частности, существует большой открытый корпус английской речи LibriSpeech [40], содержащий около 1 тыс. часов речевых данных. При этом для многих языков, называемых малоресурсными, отсутствует открытые речевые корпуса большого объема. Суть метода переноса знаний состоит в том, что вначале выполняется предварительное обучение модели на имеющихся в наличие речевых данных большого объема (их называют нецелевыми данными), а затем производится дообучение модели на имеющемся небольшом обучающем наборе речевых данных языка, для которого разрабатывается система (целевых данных). Метод переноса знаний можно выполнять по-разному. Например, можно проинициализировать параметры модели для целевой задачи параметрами, которые были получены для нецелевой задачи, и дообучить всю сеть на целевых данных. Также можно зафиксировать веса для нижних слоев нейронной сети, а дообучить только параметры средних и верхних слоев, в этом случае выполняется инициализация параметров модели для целевой задачи значениями параметров, которые были получены для нецелевой задачи, а затем при выполнении алгоритма обратного распространения ошибки обновляются параметры только средних и верхних слоев нейронной сети, при этом значения параметров нижних слоев остаются такими, какими они были в ходе предварительного обучения.

Кроме того, при недостатке обучающих речевых данных, но наличие текстовых данных в интегральных моделях можно

использовать внешнюю ЯМ, которая интегрируется на этапе декодирования следующим образом:

$$y^* = \arg\max_y \left(\log P\left(y_{hyb} \mid \mathbf{X} \right) + \psi \log P_{LM}\left(y_{hyb} \right) \right),$$

где $\mathbf{X}$ – входная последовательность признаков; y_{hyb} – выход объединенной CTC и кодер-декодер модели, ψ – вес языковой модели; P_{LM} – оценка вероятности с помощью языковой модели.

В заключение раздела следует отметить, что для создания интегральных систем распознавания речи уже существуют специально разработанные открытые программные средства, такие как EspNet [61], Eesen [34]. Также для создания интегральной системы распознавания речи могут использоваться библиотеки и программные средства для обучения нейронных сетей, например, TensorFlow, Theano, Torch, CNTK.

6.6 Вопросы по разделу 6

1. В чем состоит основное отличие интегральных систем распознавания речи от стандартных?
2. Какие модели применяются для разработки интегральных систем распознавания речи?
3. Назовите основные типы механизмов внимания.
4. Что такое мультивнимание?
5. Что такое аугментация данных? Назовите основные способы аугментации речевых данных.
6. В чем состоит суть метода переноса знаний?
7. Какие существующие программные средства для разработки интегральных систем распознавания речи вы знаете?

ЗАКЛЮЧЕНИЕ

В учебном пособии были рассмотрены основные методы автоматического распознавания речи как в рамках стандартного подхода, предполагающего независимое обучение лексических, акустических и языковых моделей, так и нового интегрального подхода, объединяющего все компоненты стандартной системы в единую нейронную сеть. Следует отметить, что большинство современных научных исследований в области автоматического распознавания речи посвящено именно интегральному подходу, однако вследствие того, что для обучения интегральных систем требуется существенно больший объем речевых данных, стандартные подходы не теряют своей актуальности.

Объем учебного пособия не позволяет осветить все возможные проблемы, возникающие при разработке систем автоматического распознавания речи. Одной из таких проблем является распознавания речи в зашумленных условиях. В этом случае могут применяться методы аудио-визуального распознавания речи. Еще одной задачей является многоязычное распознавание речи, кроме того при разработке системы распознавания речи для некоторых языков необходимо учесть наличие нескольких диалектов, а также проблему смешения языков (англ. code-switching), т.е. спонтанное переключение говорящего с одного языка на другой. Читатели, желающие более глубоко изучить подходы и методы, применяемые для автоматического распознавания речи, как освещенные в данном учебном пособии, так и оставшиеся за его рамками, могут обратиться к следующей литературе: [16, 31, 60, 64].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Amdal I.* Learning pronunciation variation. A data-driven approach to rule-based lexicon adaptation for automatic speech recognition // PhD thesis. Department of Telecommunications Norwegian University of Science and Technology. Norway, 2002.
2. *Bahdanau D., Cho K., Bengio Y.* Neural machine translation by jointly learning to align and translate // arXiv preprint arXiv:1409.0473, 2014 [Электронный ресурс]: https://arxiv.org/abs/1409.0473 (дата обращения: 20.04.2021).
3. *Bahl L.R., de Souza P.V., Gopalakrishman P.S., Nahamoo D., Picheny M.A.* Decision trees for phonological rules in continuous speech // Proceedings of ICASSP-91, Toronto, Canada, 1991. P. 185-188.
4. *Baum L.E.* An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes // Inequalities, vol.3, 1972. P. 1-8.
5. *Benesty J., Sondhi M., Huang Y. (eds.)* Springer Handbook of Speech Processing. Springer, 2008. 1176 p.
6. *Bilmes J. A., Kirchhoff K.* Factored language models and generalized parallel backoff // Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Vol. 2, Stroudsburg, PA, USA, 2003. P. 4–6.
7. *Bridle J.S.* Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition // Neurocomputing, 1990. P. 227–236.
8. *Brown P. F., Desouza P. V., Mercer R. L., Pietra V. J. D., Lai J. C.* Class-based n-gram models of natural language // Computational Linguistics, 1992, 18(4). P. 467–479.
9. *Çetinkaya G., Arısoy E., Saraçlar M.* Improving the Usage of Subword-Based Units for Turkish Speech Recognition // Proceedings of IEEE 28th Signal Processing and Communications Applications Conference (SIU), 2020. P. 1-4.
10. *Chomsky N.* On certain formal properties of grammars // Information and control, 1959, 2(2). P. 137-167.
11. *Chorowski J.K., Bahdanau D., Serdyuk D., Cho K., Bengio Y.* Attention-based models for speech recognition // Advances in Neural Information Processing Systems, 2015. P. 577–585.

12. CMU Sphinx Open Source Toolkit For Speech Recognition Evaluation [Электронный ресурс]. — Режим доступа: http://cmusphinx.sourceforge.net/ (дата обращения: 27.09.2021)
13. *Enarvi S., Smit P., Virpioja S., Kurimo M.* Automatic Speech Recognition with Very Large Conversational Finnish and Estonian Vocabularies // Proceedings of IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(11). P. 2085-2097.
14. *Fosler-Lussier E., Morgan N.* Effect of speaking rate and word frequency on pronunciations in conversational speech // Speech Communication, Vol. 29, 1999. P. 137-158.
15. *Gandhe A., Metze F., Lane I.* Neural Network Language Models for Low Resource Languages // Proceedings of INTERSPEECH-2014, 2014. P. 2615–2619.
16. Goldberg Y. Neural network methods for natural language processing // Synthesis lectures on human language technologies. 2017. Vol. 10. No 1. P. 1-309.
17. *Graves A., Ferna´ndez S., Gomez F., Schmidhuber J.* Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // Proceedings of the 23rd international conference on Machine learning, 2006. P. 369–376.
18. *Graves A., Jaitly N.* Towards end-to-end speech recognition with recurrent neural networks // Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014. P. 1764–1772.
19. Gupta D., Bansal P., Choudhary K. The state of the art of feature extraction techniques in speech recognition // Speech and language processing for human-machine communications. 2018. P. 195-207.
20. *Han H., Manavoglu E., Zha H., Tsioutsiouliklis K., Giles C. L., Zhang X.* Rule-based word clustering for document metadata extraction // Proceedings of the 2005 ACM symposium on Applied computing, 2005. P. 1049-1053.
21. *Hayashi T., Watanabe S., Toda T., Takeda K.* Multi-head decoder for end-to-end speech recognition // arXiv preprint arXiv:1804.08050, 2018 [Электронный ресурс]: https://arxiv.org/abs/1804.08050 (дата обращения: 20.04.2021).

22. *Hinton G., Deng L., Yu D., Dahl G., Mohamed A., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T., Kingsbury B.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups // IEEE Signal Processing Magazine, 2012, Vol. 29, No. 6. P. 82–97.
23. *Hirsimäki T., Pylkkönen J., Kurimo M.* Importance of High-Order N-Gram Models in Morph-Based Speech Recognition // IEEE Transactions on Audio, Speech, and Language Processing, 2009, 17(4). P. 724-732.
24. *Hruz M., Campr P., Dikici E., Kindirouglu A., Krnoul Z., Ronzhin Al., Sak H., Schorno D., Akarun L., Aran O., Karpov A., Saraclar M., Zelezny M.* Automatic Fingersign to Speech Translation System // Journal on Multimodal User Interfaces, Springer, Vol. 4, No. 2, 2011. P. 61-79.
25. *Kanungo T.* An Efficient k-Means Clustering Algorithm: Analysis and Implementation / T. Kanungo, et al // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002, vol. 24, №7. P. 881-892.
26. *Karmakar P., Teng S. W., Lu G.* Thank you for Attention: A survey on Attention-based Artificial Neural Networks for Automatic Speech Recognition // arXiv preprint arXiv:2102.07259, 2021 [Электронный ресурс]: https://arxiv.org/abs/2102.07259 (дата обращения: 20.04.2021).
27. *Karpov A., Kipyatkova I., Ronzhin A.* Very Large Vocabulary ASR for Spoken Russian with Syntactic and Morphemic Analysis // Proceedings of INTERSPEECH-2011, 2011.
28. *Kessens J. M., Wester M., Strik H.* Improving the performance of Dutch CSR by modeling within-word and cross-word pronunciation variation // Speech Communication, 1999, vol. 29. P. 193-207.
29. *Kim S., Hori T.,Watanabe S.* Joint ctc-attention based end-to-end speech recognition using multi-task learning // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2017), 2017. P. 4835-4839.
30. *Kipyatkova I.* Experimenting with Hybrid TDNN/HMM Acoustic Models for Russian Speech Recognition // Proceedings of 19th International Conference on Speech and Computer SPECOM-2017, Hatfield, UK, Springer LNCS 10458, 2017. P. 362-369.

31. *Li J., Deng L., Haeb-Umbach R., Gong Y.* Robust automatic speech recognition: a bridge to practical applications. Elsevier. 2015.
32. *Luong M. T., Pham H., Manning C. D.* Effective approaches to attention-based neural machine translation // arXiv preprint arXiv:1508.04025, 2015 [Электронный ресурс]: https://arxiv.org/abs/1508.04025 (дата обращения: 20.04.2021).
33. *Merkel A., Klakow D.* Improved Methods for Language Model Based Question Classification // Proceedings of 8th Interspeech Conference, 2007. P. 322-325.
34. *Miao Y., Gowayyed M., Metze F.* EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding // IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015. P. 167–174.
35. *Mikolov T., Kombrink S., Deoras A., Burget L., Černocký J.* RNNLM - Recurrent Neural Network Language Modeling Toolkit // Proceedings of ASRU'2011, 2011. P. 196-201.
36. *Mikolov T., Chen K., Corrado G., Dean J*. Efficient estimation of word representations in vector space // Proceedings of Workshop at ICLR, 2013.
37. *Mitchell T.* Machine Learning. New York: McGraw, 1997.
38. *Moore G.L*. Adaptive Statistical Class-based Language Modelling // PhD thesis. Cambridge University, 2001. 193 p.
39. *Oparin I., Glembek O., Burget L., Cernosky J.* Morphological random forest for language modeling of inflectional languages // Proceedings of 2nd IEEE Workshop on Spoken Language Technology, 2008. P. 189-192.
40. *Panayotov V., Chen G., Povey D., Khudanpur S.* Librispeech: an ASR corpus based on public domain audio books // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2015), 2015. P. 5206-5210.
41. *Peddinti V., Povey D., Khudanpur S.* A time delay neural network architecture for efficient modeling of long temporal contexts // Proceedings of INTERSPEECH-2015. 2015. P. 2440–2444.
42. *Pietquin O. A* framework for unsupervised learning of dialogue strategies. UCL presses, 2004. 246 p.
43. *Povey D. et al.* The Kaldi speech recognition toolkit // IEEE Workshop on Automatic Speech Recognition and Understanding ASRU, 2011.

44. *Rabiner L., Juang B.-H.* Fundamentals of Speech Recognition. Prentice Hall, 1993. 507 p.
45. *Radha V., Vimala C.* A review on speech recognition challenges and approaches // World of Computer Science and Information Technology Journal, 2012, Vol. 2, No. 1. P. 1-7.
46. *Rotovnik T., Maucec M.S., Kacix Z.* Large vocabulary continuous speech recognition of an inflected language using stems and endings // Speech Communication, Vol.49, No.6, 2007. P. 437-452.
47. SAMPA - computer readable phonetic alphabet [Электронный ресурс]. — Режим доступа: http://www.phon.ucl.ac.uk/home/sampa/ (дата обращения: 27.09.2021)
48. *Smit P., Virpioja S., Kurimo M.* Advances in subword-based HMM-DNN speech recognition across languages // Computer Speech & Language, 2021, Vol. 66, 101158.
49. *Song M., Zhao Y., Wang S.* Exploiting different word clusterings for class-based RNN language modeling in speech recognition // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2017), 2017. P. 5735-5739.
50. *Suzuki M., Itoh N., Nagano T., Kurata G., Thomas S.* Improvements to n-gram language model using text generated from neural language model // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. P. 7245-7249.
51. *Svenson M., Bhanuprasad K.* Errgrams - A Way to Improving ASR for Highly Inflective Dravidian Languages // Proceedings of 3rd International Joint Conference on Natural Language Processing IJCNLP'08, 2008. P. 805-810.
52. *Szarvas M., Furui S.* Finite-state transducer based modeling of morphosyntax with applications to Hungarian LVCSR. Proceedings of ICASSP-2003, 2003. P. 368-371.
53. *Tan L., Jiang J.* Digital signal processing: fundamentals and applications. Academic Press, 2018.
54. The Hidden Markov Model Toolkit (HTK) [Электронный ресурс]. — Режим доступа: http://htk.eng.cam.ac.uk/ (дата обращения: 27.09.2021).

55. *Vaičiūnas A.* Statistical Language Models of Lithuanian and Their Application to Very Large Vocabulary Speech Recognition // Summary of Doctoral Dissertation. Vytautas Magnus University, Kaunas, 2006, 35 p.
56. *Vaswani A. et al.* Attention is all you need // arXiv preprint arXiv:1706.03762. 2017 [Электронный ресурс]. — Режим доступа: https://arxiv.org/abs/1706.03762 (дата обращения: 27.09.2021).
57. *Vesa S., Teemu H., Mathias C., Mikko K.* Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner // Proceedings of Eurospeech, 2003. P. 2293–2296.
58. *Viterbi A.J.* Error bounds for convolutional codes and an asymmetrically optimum decoding algorithm // IEEE Transactions on Information Theory, 1967, vol. IT-13. P. 260-267.
59. *Waibel A. et al.* Phoneme recognition using time-delay neural networks // IEEE Transactions on acoustics, speech, and signal processing. 1989. Vol. 37. No. 3. P. 328–339.
60. *Wang D., Wang X., Lv S.* An overview of end-to-end automatic speech recognition // Symmetry. 2019. Vol. 11. No 8. P. 1018.
61. *Watanabe S. et al.* Espnet: End-to-end speech processing toolkit // Proceedings of INTERSPEECH-2018, 2018. P. 2207-2211.
62. *Whittaker E.W.D., Woodland P.C.* Efficient class-based language modelling for very large vocabularies. // Proceedings of ICASSP'01 Conference, 2001. P. 545-548.
63. *Young S. et al.* The HTK Book (for HTK Version 3.4). Cambridge. UK, 2009. 375 p.
64. *Yu D., Deng L.* Automatic Speech Recognition. Springer London limited, 2016.
65. *Zadeh L.* A fuzzy-algorithmic approach to the definition of complex or imprecise concepts // International Journal of Man-Machine Studies, Vol. 8, No. 3, 1976. P. 249-291.
66. *Zamora-Martínez F., Espana-Boquera S., Castro-Bleda M. J., De-Mori R.* Cache neural network language models based on long-distance dependencies for a spoken dialog system // Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP), 2012. P. 4993-4996.

67. *Zhang C., Woodland P.C.* A general artificial neural network extension for HTK // Proceedings of INTERSPEECH-2015, 2015. P. 3581–3585.
68. *Бураков М.В.* Нейронные сети и нейроконтроллеры // Учебное пособие. − СПб.: ГУАП, 2013.
69. *Гапочкин А.В.* Нейронные сети в системах распознавания речи // Science Time. 2014. № 1(1). P. 29–36.
70. *Джелинек Ф.* Распознавание непрерывной речи статистическими методами // ТИИЭР. 1976. Т. 64. № 4. С. 131-160.
71. *Заенцев И.В.* Нейронные сети: основные модели: Уч. Пособие / И.В. Заенцев. – Воронеж: Изд-во Воронежского Государственного университета, 1999, 76 с.
72. *Захаров Л. М.* Акустическая вариативность звуковых единиц в русской речи // Язык и речь: проблемы и решения: Сб. научн. трудов к юбилею проф. Л. В. Златоустовой / Под ред. Г. Е. Кедровой и В. В. Потапова. М.: МАКС-ПРЕСС, 2004. С. 240–269.
73. *Златоустова Л.В., Потапова Р.К., Трунин-Донской В.Н.* Общая и прикладная фонетика. М.: Издательство МГУ, 1986, 304с.
74. *Карпов А.А.* Модели и программная реализация распознавания русской речи на основе морфемного анализа // Диссертация на соискание ученой степени кандидата технических наук, 2007, 129 с.
75. *Карпов А.А., Кипяткова И.С., Ронжин А.Л.* Проектирование речевых интерфейсов для информационно-управляющих систем. Учеб. пособие / СПб: ГУАП. СПб., 2012, 76 с.
76. *Карпов А.А., Кипяткова И.С.* Методология количественного оценивания систем автоматического распознавания речи // Известия вузов. Приборостроение, СПб.: ИТМО, 2012, № 11, С. 38-43.
77. *Кипяткова И. С., Карпов А. А.* Модель фонематического транскрибирования для системы распознавания разговорной русской речи // Искусственный интеллект. № 4. Донецк: Украина, 2008, С. 747–757.
78. *Кипяткова И.С., Ронжин А.Л., Карпов А.А.* Автоматическая обработка разговорной русской речи. СПб.: ГУАП, 2013, 314 с.

79. *Косарев Ю.А.* Естественная форма диалога с ЭВМ. – Л.: Машиностроение, 1989, 143 с.
80. *Кривнова О. Ф.* Обработка инициальных аббревиатур при автоматическом синтезе речи // Труды международного семинара по компьютерной лингвистике и ее приложениям «Диалог 99». М., 1999, С. 4-10.
81. *Кривнова О.Ф., Захаров Л.М., Строкин Г.С.* Многофункциональный автоматический транскриптор русских текстов // Труды Международного конгресса исследователей русского языка. М., 2001, С. 408–409.
82. *Кучерявый А. А.* Бортовые информационные системы / Под ред. В. А. Мишина и Г. И. Клюева. 2-е изд., перераб. и доп. Ульяновск: УлГТУ, 2004.
83. *Маковкин К.А.* Гибридные модели – Скрытые марковские модели/Многослойный персептрон и их применение в системах распознавания речи. Обзор // Речевые технологии, 2012, № 3, С. 58–83.
84. *Марков А.А.* Исчисление вероятностей. Санкт-Петербург: Типография Императорской Академии Наук, 1913, 382 с.
85. *Марковников Н.М., Кипяткова И.С.* Аналитический обзор интегральных систем распознавания речи // Труды СПИИРАН, 2018, Вып. 58, С. 77-110.
86. *Осипов В.Ю., Никифоров В.В.* Возможности рекуррентных нейронных сетей с управляемыми элементами по восстановлению потоков кадров // Информационно-управляющие системы. – 2019. – №. 5 (102). С. 10-17.
87. *Пиотровский Р.Г.* Текст, машина, человек. – Л.: Наука, 1975, 327 с.
88. *Протасов С.В.* Вывод и оценка параметров дальнодействующей триграммной модели языка // Материалы международной конференции "Диалог 2008", Москва, 2008, С. 443-449.
89. *Прохоров А.М.* (гл. ред.) Большая советская энциклопедия. Т. 23. М.: Советская энциклопедия, 1976, 638 с.
90. *Ронжин А.Л., Карпов А.А., Ли И.В.* Речевой и многомодальный интерфейсы. - М.: Наука, 2006 - (Информатика: неограниченные возможности и возможные ограничения), 173 с.

91. *Ронжин А.Л., Ли И.В.* Методы искусственного интеллекта и автоматического распознавания речи. Учеб. пособие / СПб: ГУАП. СПб, 2006, 175 с.
92. *Светозарова Н.Д.* Некоторые особенности фонетики русской спонтанной речи // Бюллетень фонетического фонда русского языка №8, Фонетические свойства русской спонтанной речи. СПб: Бохум, 2000, С. 7-15.
93. *Скрелин П. А.* Сегментация и транскрипция. СПб, 1999.
94. *Соловьева Е. Б.* Рекуррентные нейронные сети в качестве моделей нелинейных динамических систем // Цифровая обработка сигналов. 2018. №1. С. 18-27.
95. *Тампель И. Б., Карпов А. А.* Автоматическое распознавание речи //Учебное пособие.− СПб: Университет ИТМО, 2016.
96. *Шведова Н. Ю.* (гл. ред.). Русская грамматика: [В 2 т.]. М.: Наука, 1980.
97. *Шеннон К.* Работы по теории информации и кибернетике. — М.: Изд. иностр. лит., 2002
98. *Щерба Л. В.* Языковая система и речевая деятельность. Л., 1974.

Учебное издание

Кипяткова Ирина Сергеевна
Карпов Алексей Анатольевич
Кулешов Сергей Викторович
Зайцева Александра Алексеевна

МЕТОДЫ И МОДЕЛИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ

Учебное пособие

ISBN 978-5-6047036-0-1

Подписано к печати 01.11.21. Формат 60x84 1/16.
Усл. печ. л. 6,45. Уч.-изд. л. 6,93. Тираж 250 экз. Заказ №

Отпечатано с оригинал-макета СПб ФИЦ РАН
199178, Санкт-Петербург, 14-я линия В.О., д. 39
в редакционно-издательском центре ГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67