

ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
Центр высшего образования

Кафедра математических и информационных технологий

Беляев Станислав Валерьевич

**TalkNet: конволовционная
неавторегрессионная генерация речи**

Магистерская диссертация

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Омельченко А. В.

Научный руководитель:
д. ф.-м. н., профессор Омельченко А. В.

Рецензент:
PhD, Гинзбург Б. Е.

Санкт-Петербург
2020

Оглавление

Введение	3
1. Обзор предметной области	5
1.1. Постановка задачи	5
1.2. Особенности предлагаемого подхода	6
1.3. Обзор методов генерации речи	8
2. Описание подхода	16
2.1. Извлечение истинных длительностей графем	16
2.2. Предсказатель длительностей графем	20
2.3. Генератор мэл-спектрограмм	21
3. Решение	26
3.1. Данные для обучения	26
3.2. Обучение предсказателя длительностей графем	27
3.3. Обучение генератора мэл-спектрограмм	28
4. Результаты	31
4.1. Качество аудио	31
4.2. Скорость генерации	32
Заключение	36
Список литературы	38

Введение

Нейронные сети и глубокое обучение позволили достичнуть результатов, сравнимых с теми, что может показывать человек, на различных задачах машинного обучения. Глубокие сети с residual соединениями [6] позволили обогнать человека на задаче классификации изображений. Трансформеры [2] позволили научиться лучше понимать естественный язык и генерировать длинные отрывки текста подобно человеку. Генеративные состязательные сети [12] (GAN) позволили научиться генерировать изображения, не отличимые от оригинальных.

Однако, некоторые задачи остаются нерешенными до сих пор. Например, для задачи генерации аудио с речью по текстовому отрывку до сих пор не удается достичнуть качества голоса, не отличимого от записи настоящей человеческой речи. Более того, большинство подходов в области генерации речи требуют большое количество параметров и долго обучаются, что делает их труднодоступными для использования. Одна из дополнительных насущных проблем современных подходов – это поддержание быстрой скорости генерации для возможности использования в realtime системах. Однако простые рекуррентные авторегрессионные подходы сильно ограничивают скорость с которой можно делать вывод.

Генерация речи имеет множество применений в современной жизни. Программы для синтеза используют повсеместно для голосовых помощников, перевода текстовой информации в аудио и систем для помощи инвалидам.

Таким образом, целью данной работы является разработка модели машинного обучения, позволяющей производить эффективную, качественную и быструю генерацию речи из входного текста. В рамках данной работы, решение будет основываться на сверточных (конволюционных) сетях с неавторегрессионной архитектурой, позволяющей рассчитывать на максимальную производительность на современных графических ускорителях (GPU).

Для достижения описанной выше цели необходимо решить следующие задачи:

- Проанализировать предметную область и существующие модели. Обозначить основные проблемы и пути к их решению.
- Разработать и описать эффективную архитектуру, основанную на идеи неавторегрессионности.
- Выбрать данные для обучения и провести эксперименты.
- Произвести сравнение подходов и анализ результатов на качество и скорость.

В рамках данной работы предлагается TalkNet: сверточная неавторегрессионная нейронная модель, которая решает задачу синтеза речи. Модель состоит из двух прямых (feed-forward) полностью сверточных нейронных сетей. Первая сеть служит

для предсказания длительности входных символов (графем), выравнивая таким образом входную последовательность на длину мэл-спектrogramмы. Далее, производится операция расширения (expansion) входного текста путем повторения каждого символа в соответствии с предсказанной длительностью. Вторая сеть генерирует мэл-спектrogramму из развернутого текста. Операция разворачивания, таким образом, позволяет построить неавторегрессионную архитектуру.

Чтобы обучить предиктор длительностей графем, истинные длительности для набора данных для обучения получаются из предварительно обученной модели для распознавания речи на основе Connectionist Temporal Classification (CTC) функции ошибки. Явное предсказание длительностей исключает пропуски и повторения слов. Эксперименты с набором данных LJSpeech показывают, что качество речи TalkNet сравнимо с авторегрессионными подходами. Модель очень компактна – она имеет всего около 10,8 миллионов обучаемых параметров, что почти в 3 раза меньше, чем предлагают современные модели преобразования текста в речь. Неавторегрессионная архитектура также позволяет быстро обучаться и делать выводы.

В главе 1 будут описана формальная постановка задачи генерации речи и существующие подходы к ее решению, а также их недостатки и достоинства. В главе 2 происходит описание главной идеи TalkNet с разбиением процесса генерации на два шага. Глава 3 посвящена описанию данных и проводимых экспериментов, а также подробностям процесса обучения. В главе 4 проводится анализ результатов с точки зрения качества и скорости.

1. Обзор предметной области

1.1. Постановка задачи

Задача генерации речи имеет простую математическую формулировку. По заданному отрывку текста со строковым представлением (конечная последовательность символов из конечного алфавита), сгенерировать аудиодорожку с хорошо слыши- мой человеческой речью соответствующей заданному отрывку. Формат аудио обычно представляет из себя последовательность из 16-битных действительных чисел от -1.0 до 1.0 – дискретное приближение непрерывной аудиоволны. Основной параметр такого формата - это частота дискретизации, выражаемая в герцах (Гц, Hz). Популярными частотами для аудио отрывков в системах генерации речи являются 22.05 кГц и 24 кГц (22050 и 24000 значения в секунду соответственно). Чем выше частота дискретизации – тем лучше качество аудио (так как лучше приближение), но тем сложнее уловить зависимости для генерации (Рисунок 1).

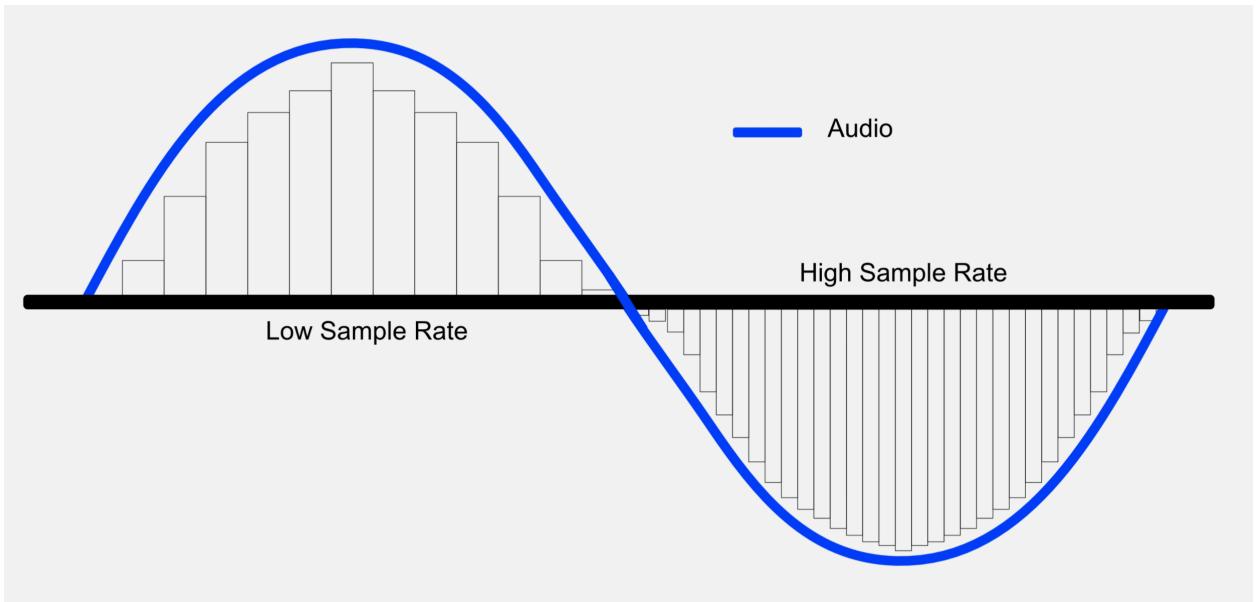


Рис. 1: Примеры различной частоты дискретизации (sample rate) для представления аудио

Модели на основе нейронных сетей (NN) для преобразования текста в речь (Text-To-Speech, TTS) превзошли как конкатенативный (concatenative), так и статистический параметрический подходы для синтеза речи с точки зрения качества. Они также значительно упрощают процесс синтеза речи. Традиционно. системы синтеза речи объединяют несколько блоков: модель для извлечения лингвистических признаков из текста, модель предсказания длительности, модель предсказания акустических признаков и вокодер на основе обработки сигналов [27], который служит для преобразования акустических признаков в аудиодорожку. Нейронные TTS системы, как правило, имеют два этапа (Рисунок 2). На первом этапе модель генерирует мэл-спектограммы

из текста. На втором этапе вокодер на основе нейронной сети синтезирует речь из мэл-спектрограмм. Большинство моделей TTS на основе нейронных сетей имеют архитектуру encoder-decoder [3] с операциями с механизмом внимания (attention), которые, как было замечено, имеют некоторые общие проблемы:

1. Тенденция повторять или пропускать слова [11], из-за сбоев работы механизма внимания, когда некоторые подпоследовательности повторяются или игнорируются. Для решения этой проблемы модели, основанные на механизме внимания, используют дополнительные правила поощрения монотонного внимания [19, 8, 28]. В общем же случае, механизм внимания часто ведет к проблемам на этапа вывода, а также замедляет скорость обучения, так как обычно состоит как минимум из одной операции с квадратичной по времени асимптотикой.
2. Медленная скорость вывода относительно параметрических моделей.
3. Нет простого способа контролировать просодию (паттерн ритма и интонации голоса) или скорость голоса, так как длина генерируемой последовательности определяется декодером.

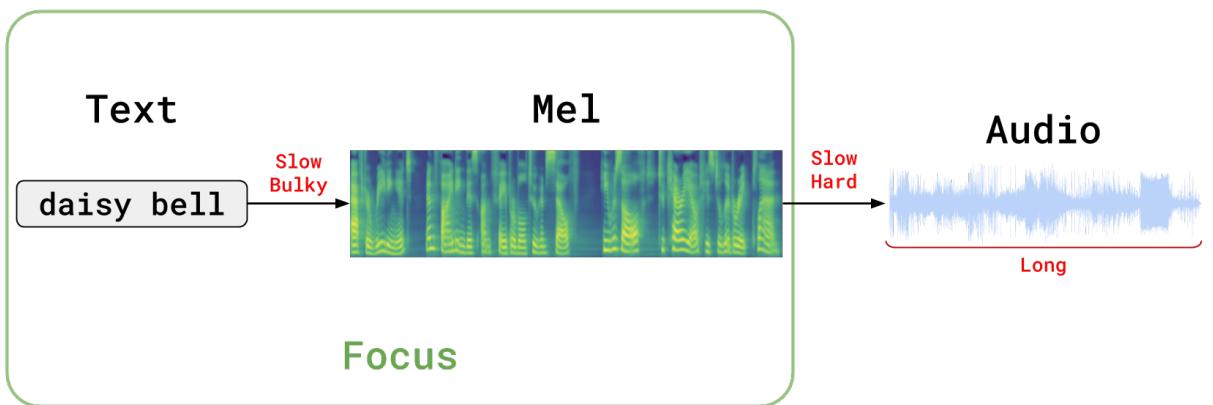


Рис. 2: Два шага TTS систем. Первый шаг – генерация мэл-спектрограммы – будет рассматриваться в рамках данной работы. Второй шаг – вокодинг – отдельная сложная задача. Оба шага имеют полно нерешенных проблем: медленная скорость работы, большое количество весов, заметно уступающее качество в сравнении с записью человеческой речи.

1.2. Особенности предлагаемого подхода

В рамках данной работы описывается новая нейронная модель TTS для решения проблем описанных выше. Модель состоит из двух сверточных сетей. Первая сеть предсказывает длительности графем (входных символов). Далее, входной текст расширяется, повторяя каждый символ в соответствии с предсказанной длительностью.

Вторая сеть генерирует мэл-спектограммы из развернутого текста. Наконец, используется вокодер WaveGlow [23] для синтеза звука из мэл-спектограмм (Рисунок 3). Финальная часть – вокодер – не является частью модели и должна рассматриваться в рамках отдельной задачи, но в данной работе будет использоваться WaveGlow из-за наличия универсальной претренированной модели и возможности сравнения с другими подходами.

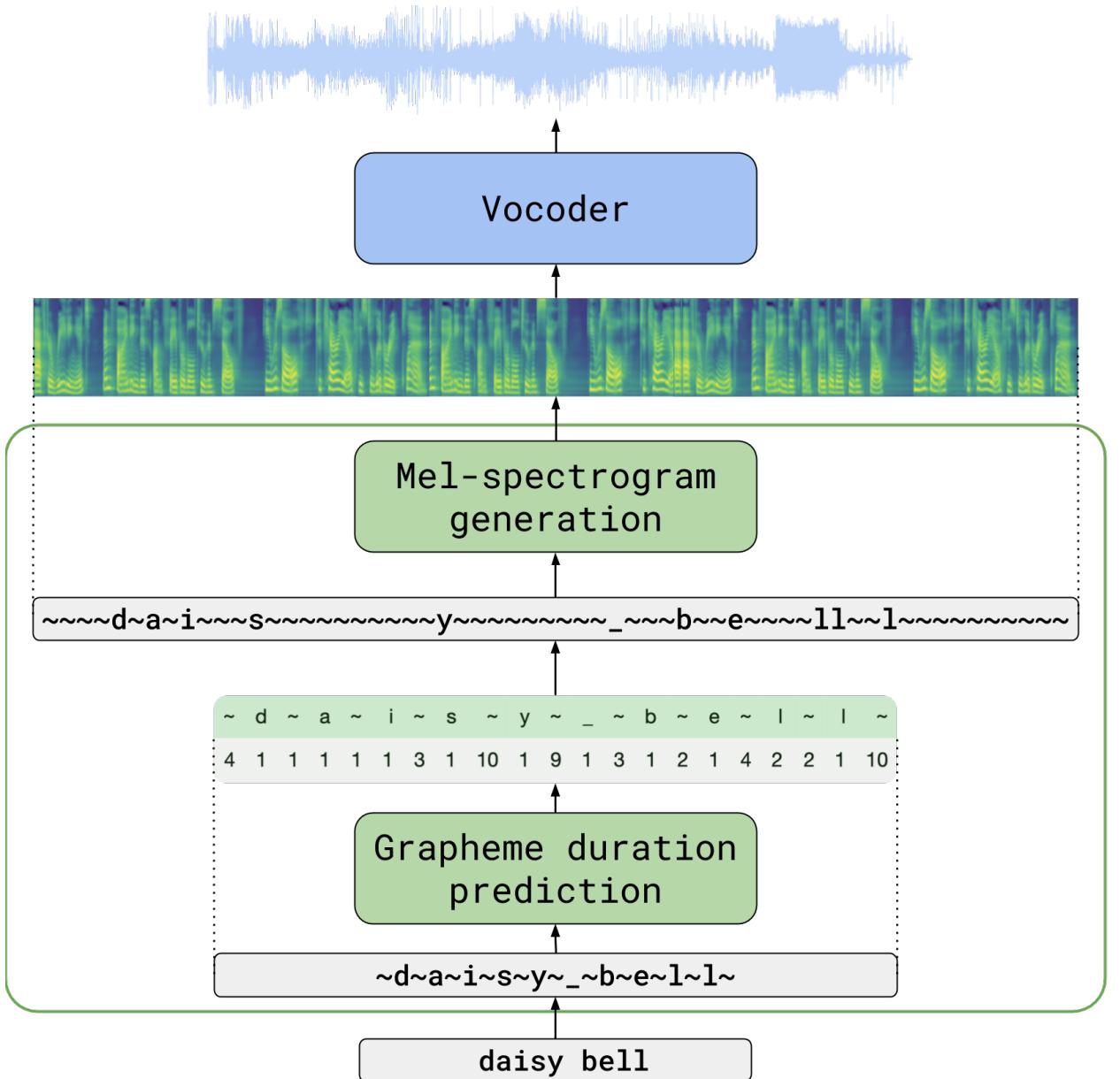


Рис. 3: TalkNet преобразует текст в речь, используя предиктор длительностей графем, генератор мэл-спектрограмм и вокодер. \sim используется в рамках данной работы для обозначения пустого символа из выхода СТС функции ошибки.

Чтобы обучить предсказатель длительностей графем, нам нужно сначала получить истинное выравнивание между входными символами и звуковой дорожкой по времени. Аналогичная проблема выравнивания существует в автоматическом рас-

познавании речи (ASR), которая решается явным образом с помощью Connectionist Temporal Classification (CTC) [13]. СТС маргинализирует вывод по всем возможным выравниваниям, выбирая наилучшее. Выбирая наиболее вероятный выход в каждый момент времени, его можно использовать его для выравнивания между входным звуком и выходным текстом. Это выравнивание не является совершенным, и в нем могут быть ошибки. В рамках данной работы показано, что если модель ASR точна и имеет низкую частоту ошибок в символах (Char-Error-Rate, CER), то можно извлечь достаточно хорошое выравнивание между текстом и звуком. Полученное выравнивание на основе СТС можно применить для обучения модели, которая будет предсказывать длительности графем для входного текста. Предиктор длительности графемы заменяет выравнивание на основе механизма внимания (attention) и позволяет избежать пропуски и повторения слов. При этом эксперименты с набором данных LJSpeech [15] показывают, что качество речи для TalkNet сравнимо с лучшими авторегрессионными подходами.

Конволюционная структура обоих частей позволяет проводить параллельное обучение, также значительно ускоряет скорость вывода. Такая структура позволяет работать значительно быстрее со значительно меньшим числом параметров, поддерживая качество генерируемой речи аналогичное FastSpeech [11] и Tacotron 2 [19].

1.3. Обзор методов генерации речи

Методы синтеза речи, основанные на статистике, обычно имеют следующие части: преобразователь графем в фонемы, предиктор длительностей фонемы, генератор акустических признаков (например, мэл-спектрограмм) и вокодер [27]. Zen et al [14, 29, 10] предложили гибридную нейронную параметрическую модель TTS (Рисунок 4 и 5), в которой глубокие нейронные сети используются для прогнозирования длительности фонемы и генерации акустических признаков на уровне кадра. Предиктор длительности фонемы был обучен на основе скрытой марковской модели (HMM), из которой извлекались фонетические выравнивания.

Один из других возможных подходов – DeepVoice [9, 7] (Рисунок 6 и 7) – также основан на для традиционной для области синтеза речи структуре, но заменяет обучаемые компоненты на нейронные сети. Для обучения предиктора длительности фонем была использована вспомогательная модель на основе СТС для фонетической сегментации, позволяющая аннотировать данные по границам фонем. Другая модель – Tacotron [26, 19] (Рисунок 8) – это end-to-end нейронная сеть, которая принимает символы в качестве входных данных и сразу выводит мэл-спектрограмму слева направо шаг за шагом (авторегрессионно). Tacotron 2 использует архитектуру encoder-decoder с механизмами внимания. Encoder состоит из трех сверточных слоев и одного двунаправленного LSTM. Decoder представляет собой рекуррентную нейронную

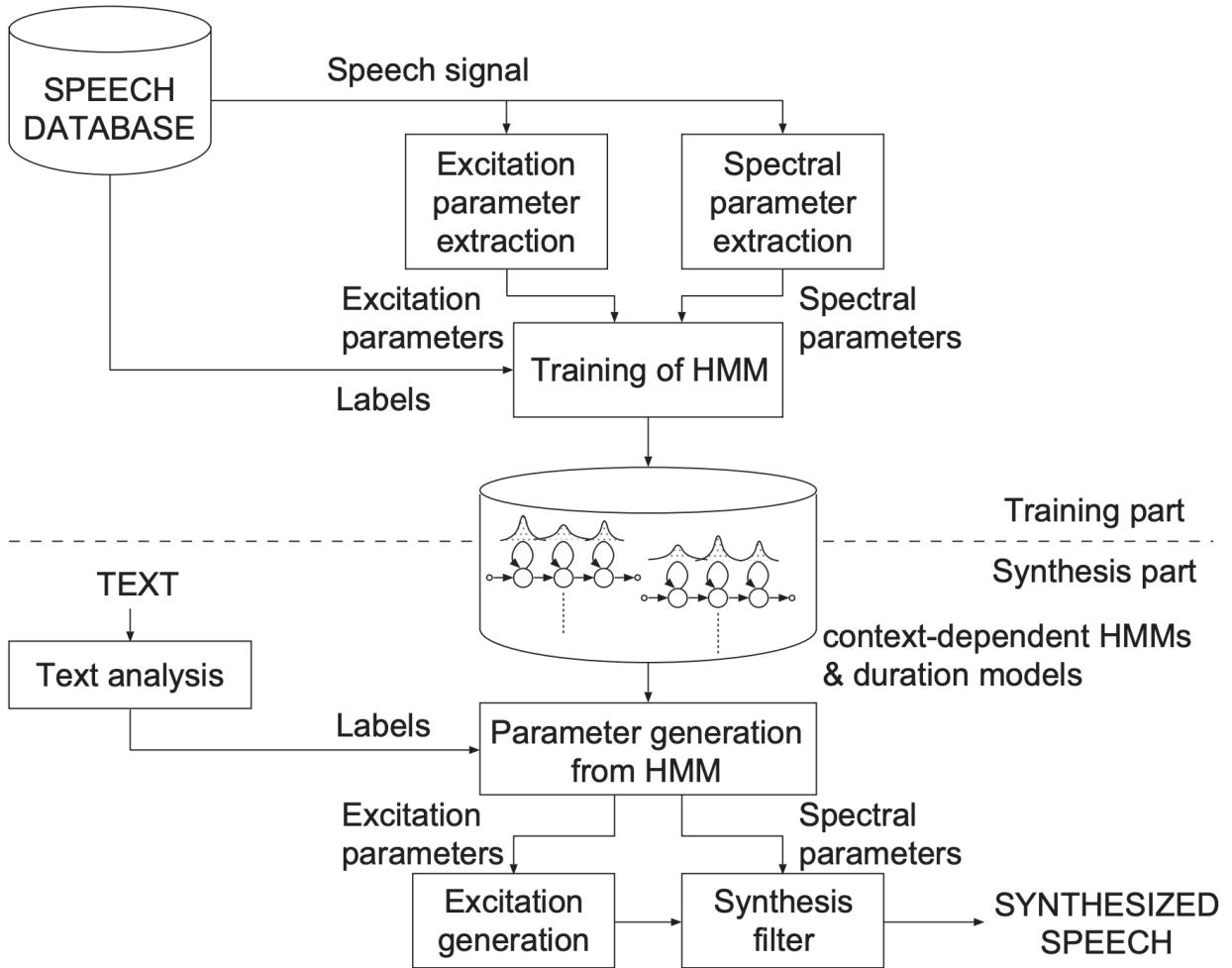


Рис. 4: Блок-диаграмма работы статистической ТТС системы с предсказанием длительностей графем [14]

сеть (RNN) с чувствительным к местоположению монотонным вниманием (location-sensitive monotonic attention). Авторегрессионность Tacotron2 позволяет хорошо учить контекст, а механизмы внимания позволяют правильно расставлять акценты при произношении. Tacotron2 – одна из лучших на сегодняшний день моделей по качеству, однако для ее обучения требуется много времени (до нескольких дней).

Последовательный характер моделей на основе рекуррентных нейронных сетей (RNN) ограничивает эффективность обучения и вывода. Но для генерации речи не обязательно использовать RNN. DeepVoice 3 [8] (Рисунок 9) заменяет RNN на конволовационную модель с encoder-decoder архитектурой и монотонным механизмом внимания. Переход от RNN к сверточной нейронной сети (CNN) ускоряет обучение, но вывод модели по-прежнему является авторегрессионным. Другой end-to-end моделью TTS, которая не использует RNN, является ParaNet [22] (Рисунок 10). ParaNet – это еще одна конволовационная encoder-decoder с механизмом внимания. Для обучения ParaNet требуется другая предобученная TTS модель, у которой заимствуется матрица внимания. Наконец, Transformer-TTS и [21] (Рисунок 11) заменяет рекур-

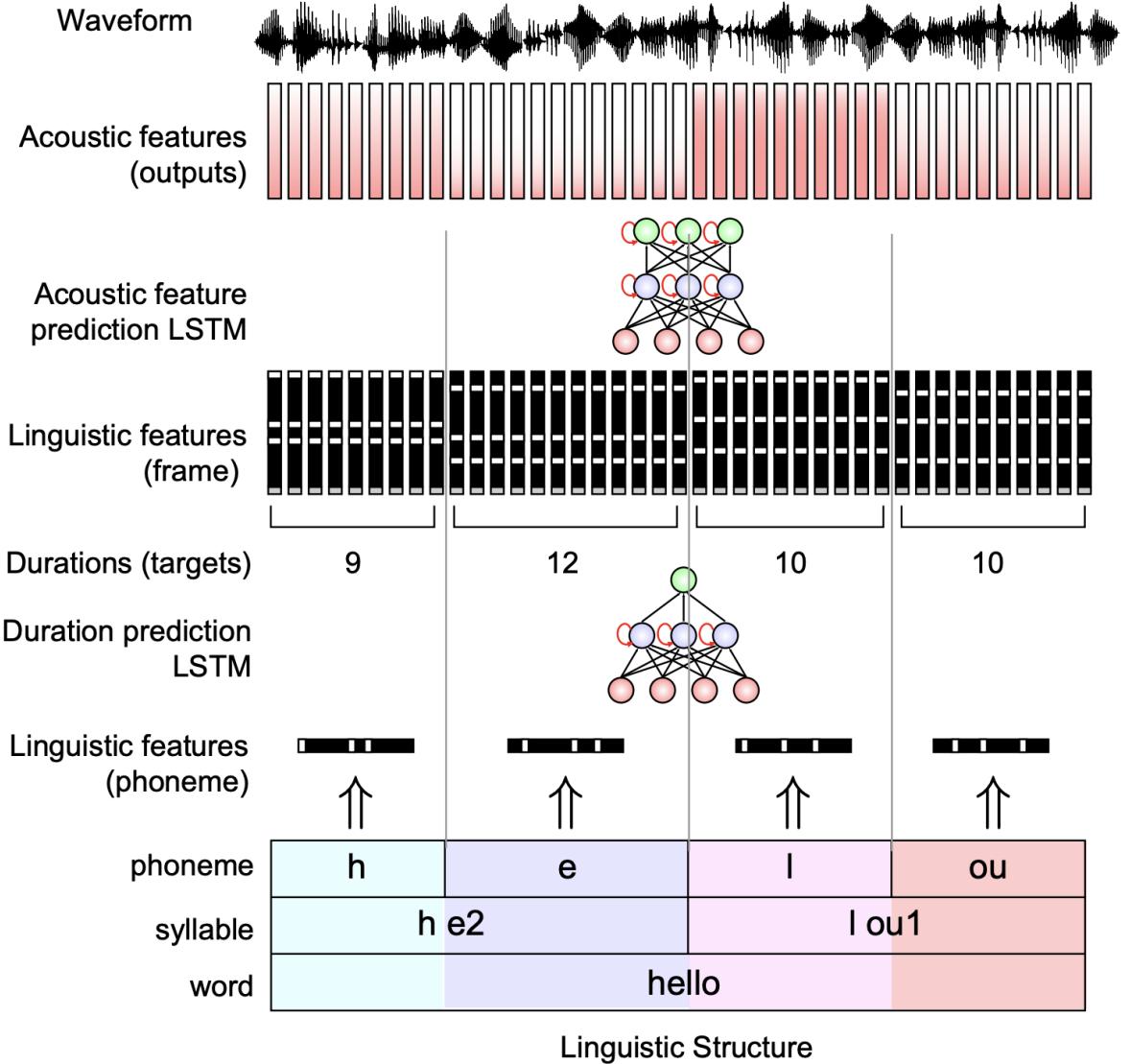


Рис. 5: Более современный вариант модели с предсказыванием длительностей входных графем с использованием нейронных сеток [29, 10]

рентные нейронные сети с encoder-decoder структурой на трансформеры [2] (модель, основанная на применении механизма внимания несколько раз подряд). Transformer-TTS сначала преобразует текст в фонемы с помощью конвертера на основе правил. Используя последовательности фонем в качестве входных данных, Transformer-TTS генерирует мэл-спектрограмму.

Как и в других моделях, основанных на внимании, Tacotron, Transformer-TTS и ParaNet иногда пропускают или повторяют слова [22]. Чтобы предотвратить пропуск и повторение слов, FastSpeech [11] предлагает end-to-end модель с использованием трансформеров [2] взамен традиционной структуре encoder-attention-decoder. FastSpeech использует явный регулятор длины в виде отдельного декодера, который расширяет последовательность фонем в соответствии с предсказанной длительностью.

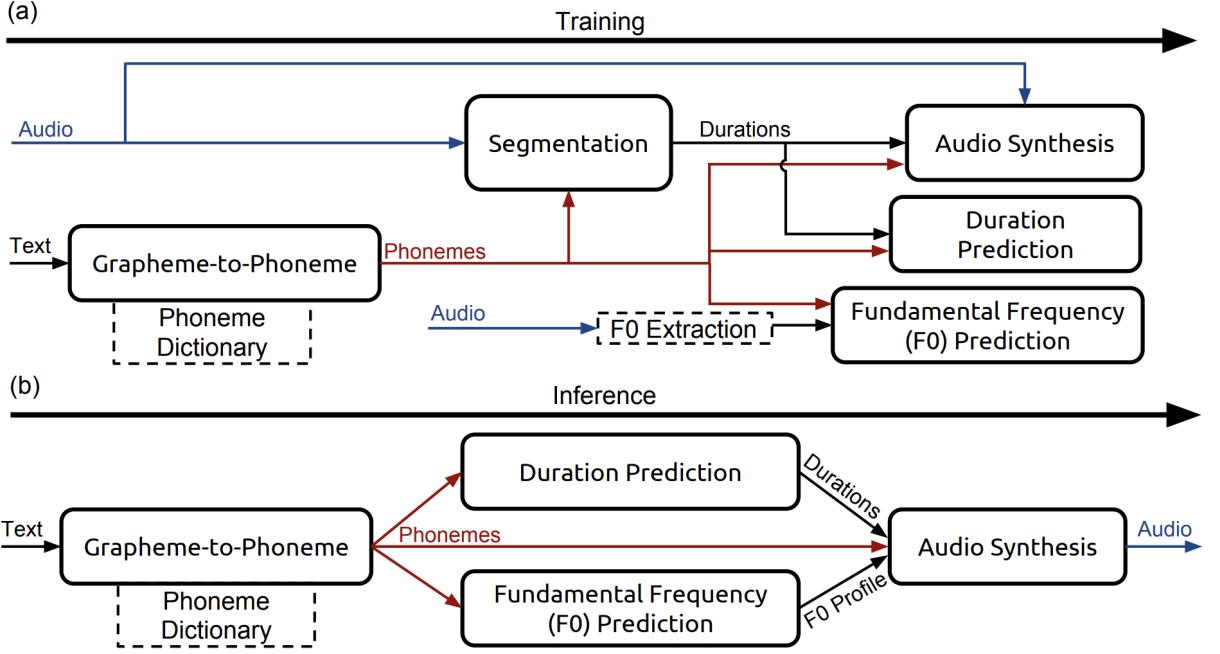


Рис. 6: Диаграмма описывающая модель DeepVoice1 [9] с промежуточным шагов в виде предсказания длительностей

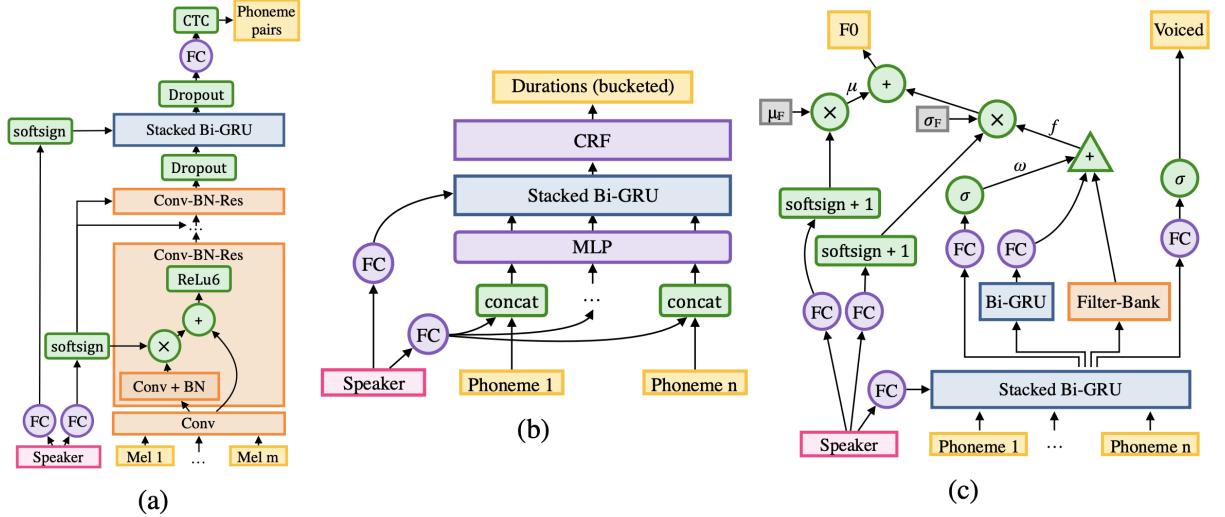


Рис. 7: DeepVoice2 с multi-speaker расширением

ностью, чтобы длина последовательности стала соответствовать длине мэл-спектрограммы. Длительности фонем извлекаются из выравнивания механизма внимания во внешней предварительно обученной модели TTS, Tacotron 2. Такой подход показывает неплохие результаты с точки зрения качества и значительно повышает скорость генерации. FastSpeech – первая модель, показавшая, что идею с промежуточным шагов предсказания длительностей можно успешно применять для построения хорошо работающей TTS системы. Во многом, результаты текущей работы, основаны на про-

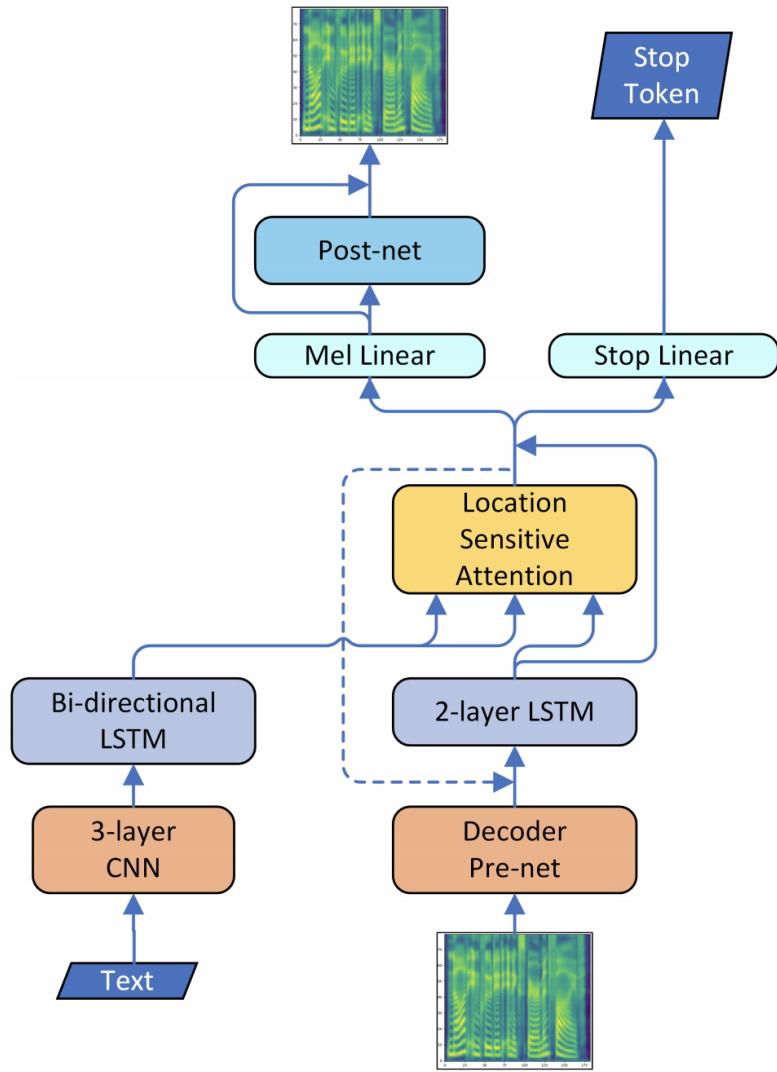


Рис. 8: Процесс генерации речи для авторегрессионного Tacotron2

должении и развитии идей FastSpeech.

Как видно, идеи из области глубокого обучения позволили упростить подходы для генерации речи.

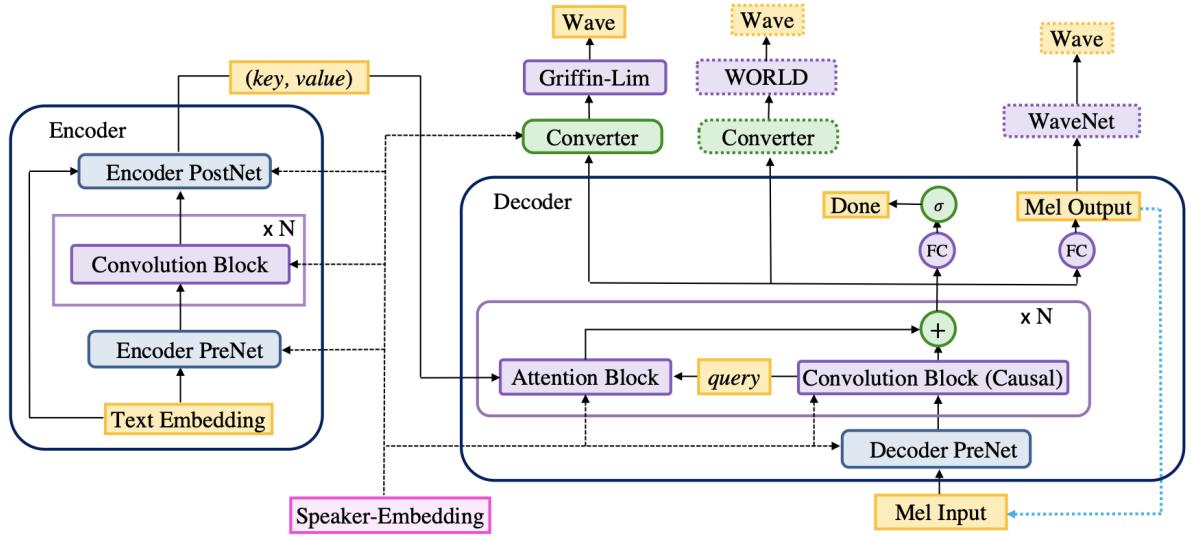


Рис. 9: Encoder-decoder структура в архитектуре DeepVoice3

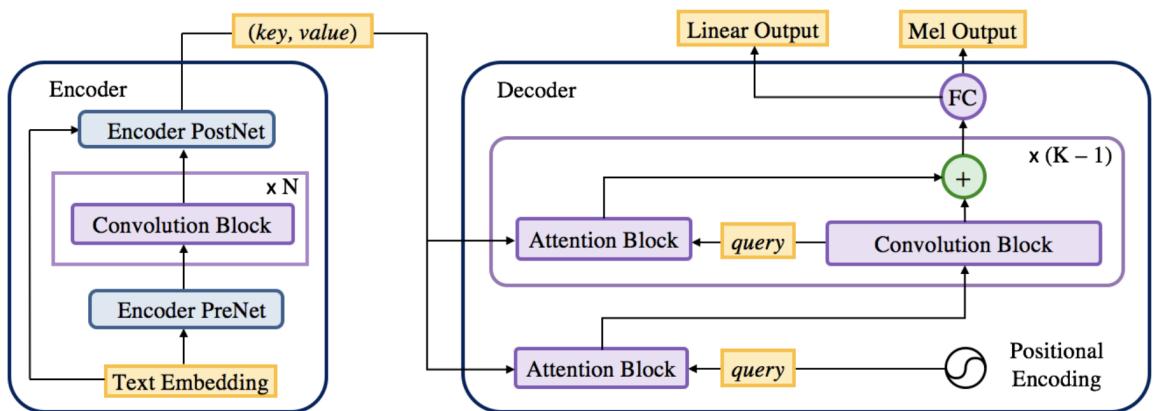


Рис. 10: Архитектура ParaNet [22]

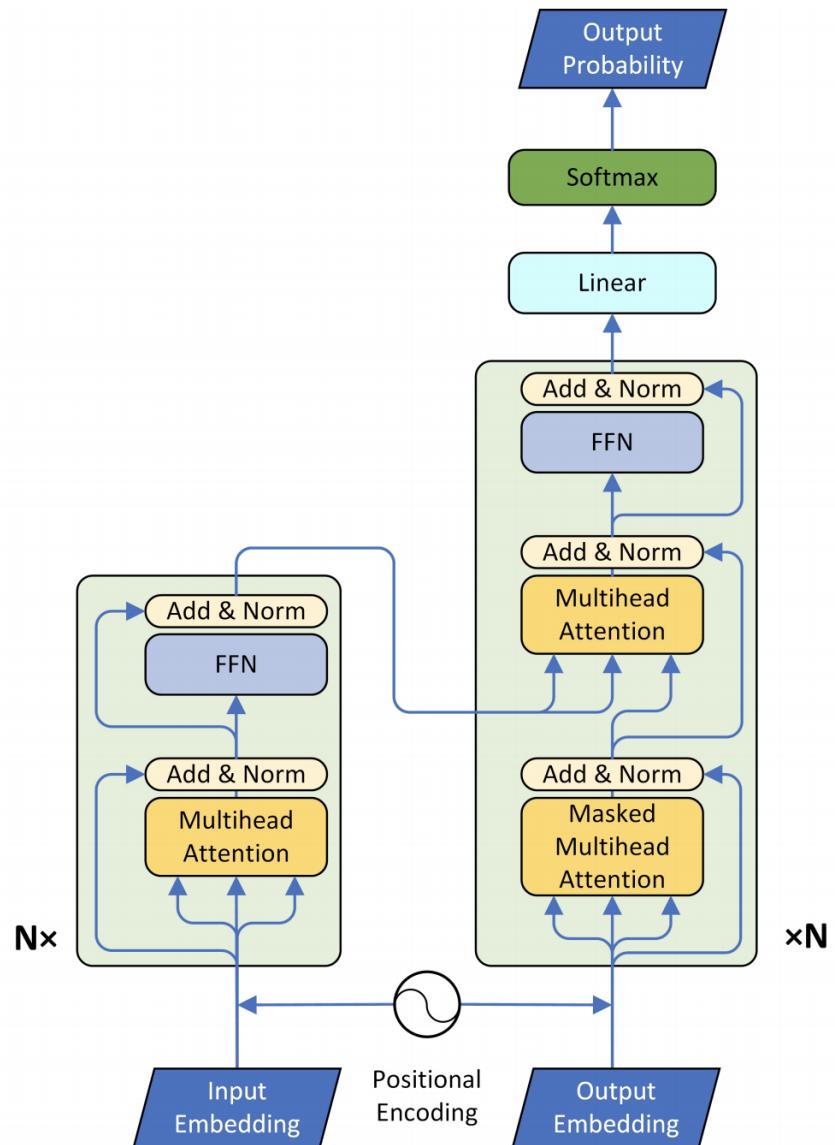


Рис. 11: Encoder-decoder Архитектура Transformer-TTS [21]

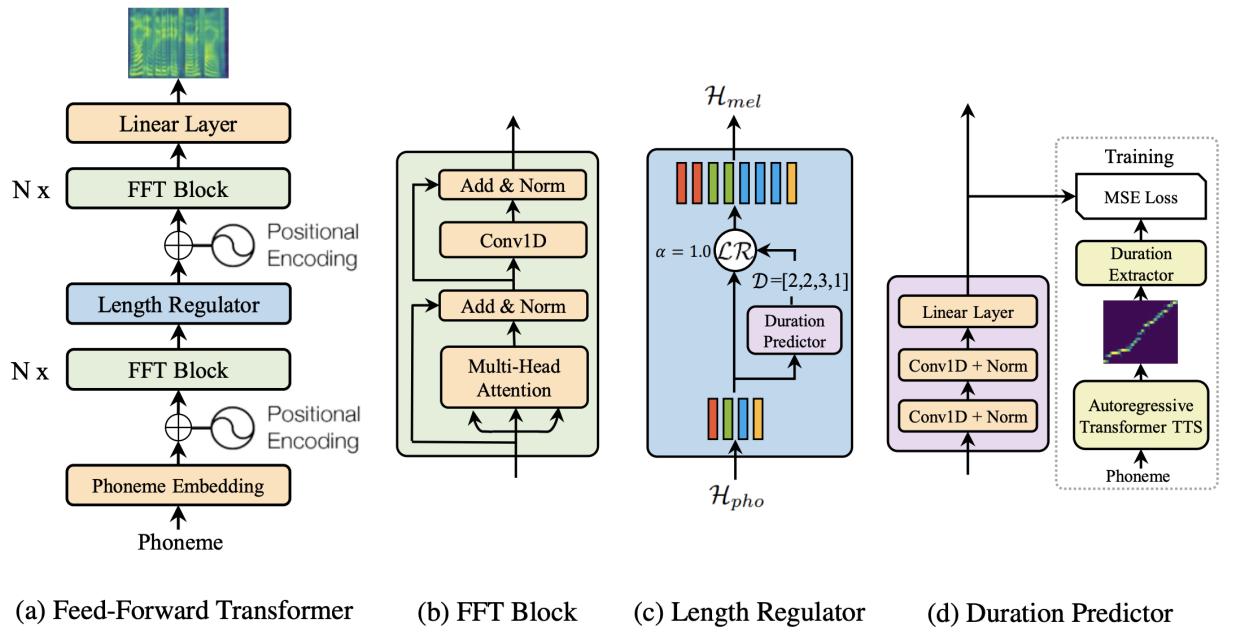


Рис. 12: Архитектура FastSpeech [11]

2. Описание подхода

TalkNet разбивает генерацию мэл-спектrogramмы из текста на два отдельных модуля. Первый модуль, предсказатель длительности, выравнивает входные графемы по времени относительно звуковой дорожки (или мэл-спектrogramмы, что то же самое так как длина мэл-спектrogramмы линейно зависит от длины аудио). Второй модуль, генератор мэл-спектrogramм, производит генерацию из выровненных по времени входных символов (Рисунок 3). Были использованы конволюционные модели прямого вывода (feed-forward) для обоих модулей, поэтому как обучение, так и вывод не являются авторегрессионными. Это позволяет гораздо быстрее обучаться и делать вывод по сравнению с авторегрессионными подходами. Для обучения предсказателя, истинные длительности графем извлекались из выхода СТС для предварительно обученной модели распознавания речи (Automatic-Speech-Recognition, ASR).

Таким образом, вводя дополнительный шаг на пути к получению мэл-спектrogramм, появляется возможность явным образом контролировать манеру произношения (просодию). Длительности букв можно изменить вручную, указав где нужно сделать паузу, а где наоборот проговорить быстро. Обе части могут обучаться независимо и параллельно, поэтому такой переход от end-to-end архитектуре к нескольким шагам оправдан с точки зрения времени.

2.1. Извлечение истинных длительностей графем

Центральная идея TalkNet заключается в использовании модели ASR на основе Connectionist Temporal Classification (CTC) функции ошибки для извлечения выравниваний графем. СТС присваивает вероятность каждому из символов алфавита и использует вспомогательный пустой символ \sim . Первым шагом схлопываются соседние повторяющиеся символы в выводе, подсчитывая таким образом длительность каждого символа. Пустой символ выступает как промежуточное состояние между двумя соседними графемами, и его длительность соответствует длительности перехода от одного символа к другому. Для каждого временного шага выбирается наиболее вероятный символ из выходных данных СТС (Рисунок 13).

СТС – это функция ошибки, используемая для этапа обучения. Поэтому, выход СТС часто бывает неточен. Однако, задача ASR решена намного лучше TTS, поэтому ошибка все еще намного меньше. Выход СТС выравнивается с истинным отрывком текста для того чтобы максимально устранить влияние ошибки. Была использована функция *pairwise2* из пакета Biopython [5] (пакет для использования в биоинформатике для языка Python), которая выравнивает два строковых представления побуквенно, используя наименьшее количество операции добавления и удаления символов. Затем удаляются все неправильные символы в выводе СТС, а их длительность добавляется к ближайшему пустому, а также добавляются недостающие символы с



Рис. 13: Пример работы СТС алгоритма [13]. Соседние буквы схлопываются, а ϵ (~ в нашей нотации) служит разделительным вспомогательным символом.

длительность 0. Затем, все символы с предсказанной длительностью 0 получают длительность 1, вычитая 1 из почти самого большого соседнего \sim , чтобы сумма всех длительностей графемы была равна длине мэл-спектограммы (Рисунок 14).

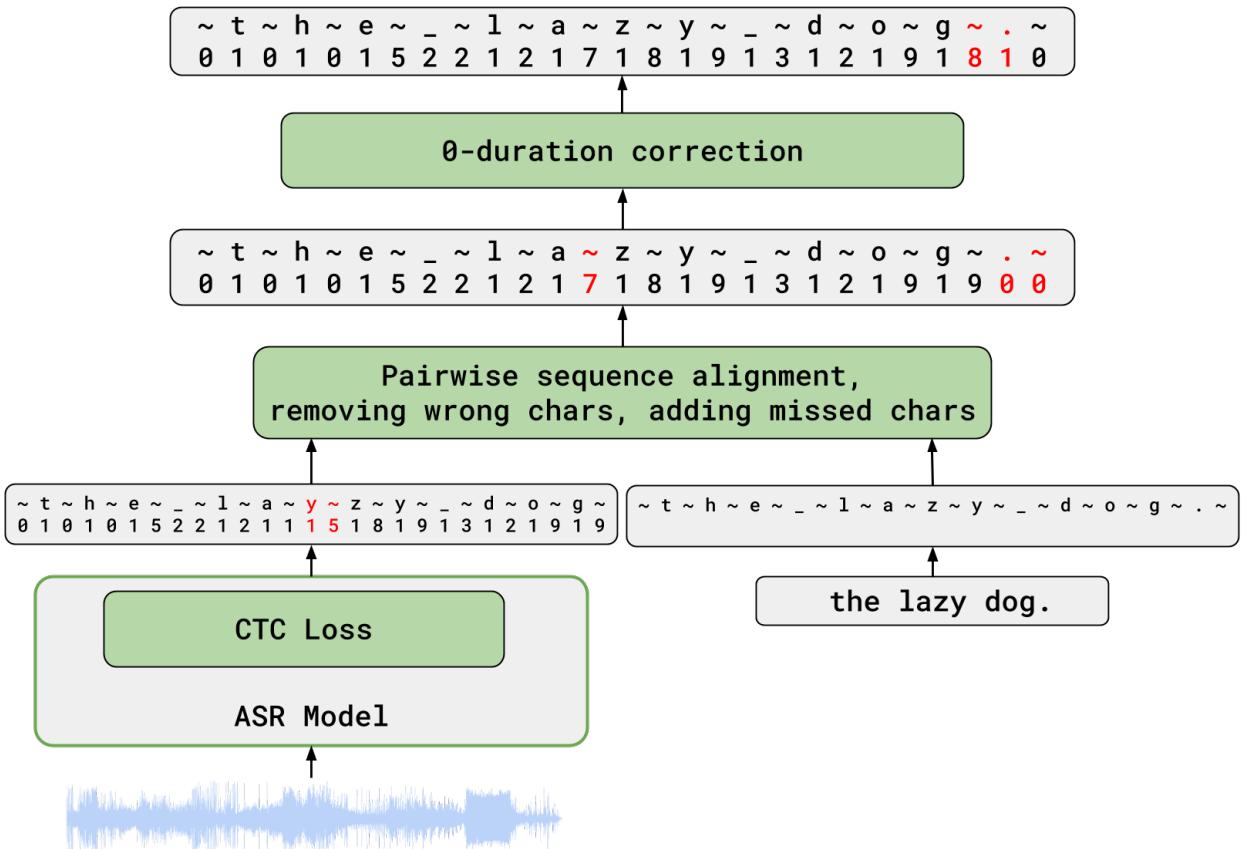


Рис. 14: Извлечение длительности графемы из вывода СТС. \sim используется для обозначения пустого символа в СТС.

В качестве модели с СТС выводом для задачи распознавания текста (ASR) ис-

пользуется QuartzNet [24]. QuartzNet (Рисунок 16) – это полностью ковалюционная нейронная архитектура, основными достоинствами которой являются:

- Низкое количество параметров (около 18 миллионов), которое было достигнуто за счет использования depthwise separable [16] конволюций, являющихся математическим приближение обычных конволюций (Рисунок 15).
- Простая неавторегрессионная архитектура с базовыми операциями из глубокого обучения (конволюции, нелинейности, батч-нормализация и дропаут), позволяющая значительно ускорить процесс обучения и вывода (inference).
- СТС функция в качестве функции ошибки в декодере, которая не содержит дополнительных параметров и обеспечивает быстрый вывод.

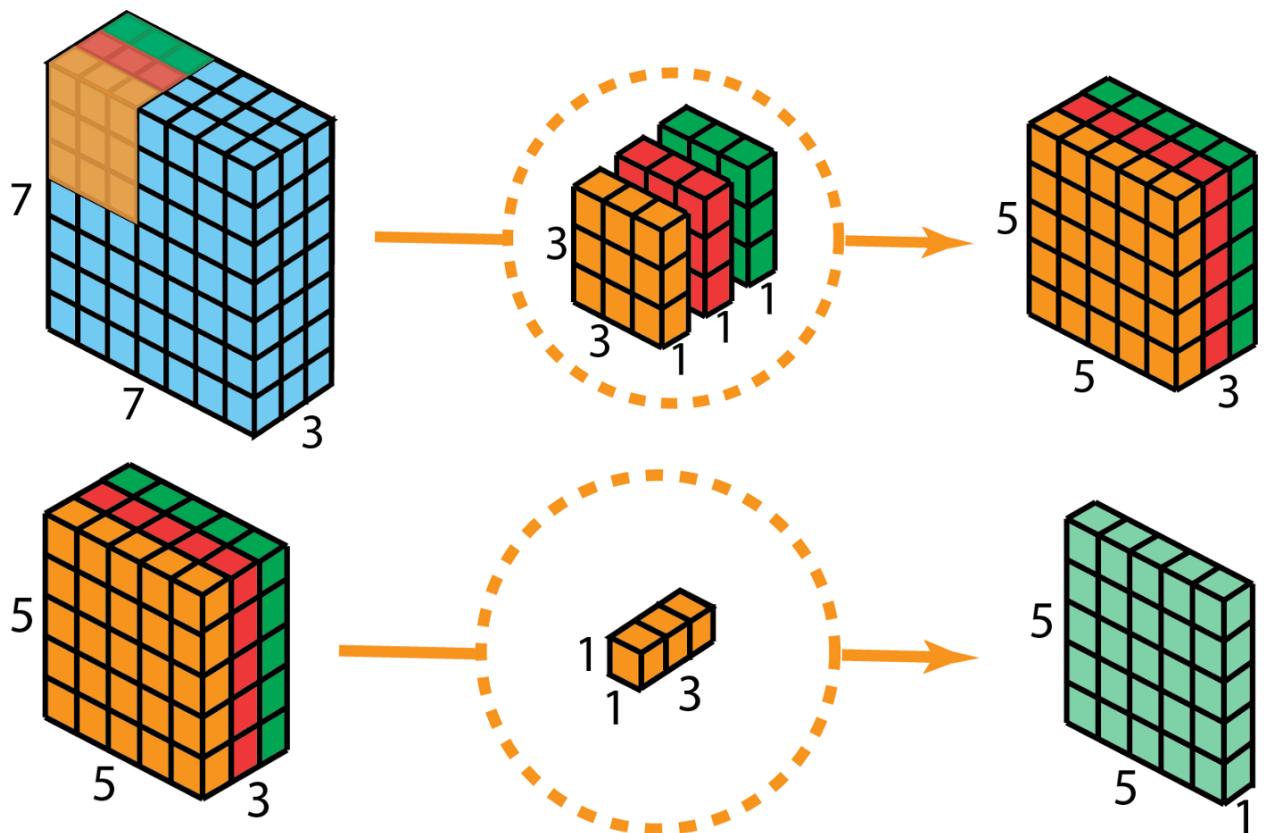


Рис. 15: Depthwise separable конволюции. Применяется в два этапа: на первом используется 1d конволюции по времени, на втором применяются 1×1 pointwise свертки. Два шага работают сообща и действуют как аппроксимация обычных сверток к квадратичными ядрами. В TalkNet такие свертки реализованы напрямую, через две последовательные конволюции.

Для получения истинных длительностей графем была использована архитектура QuartzNet 15x5 (15 блоков по 5 повторений). Выход такой модели по длине в 2 раза меньше входной мэл-спектрограммы. Причина – в самой первой конволюции выставлен параметр **stride = 2** (Рисунок 17). QuartzNet использует удвоение шага в самом

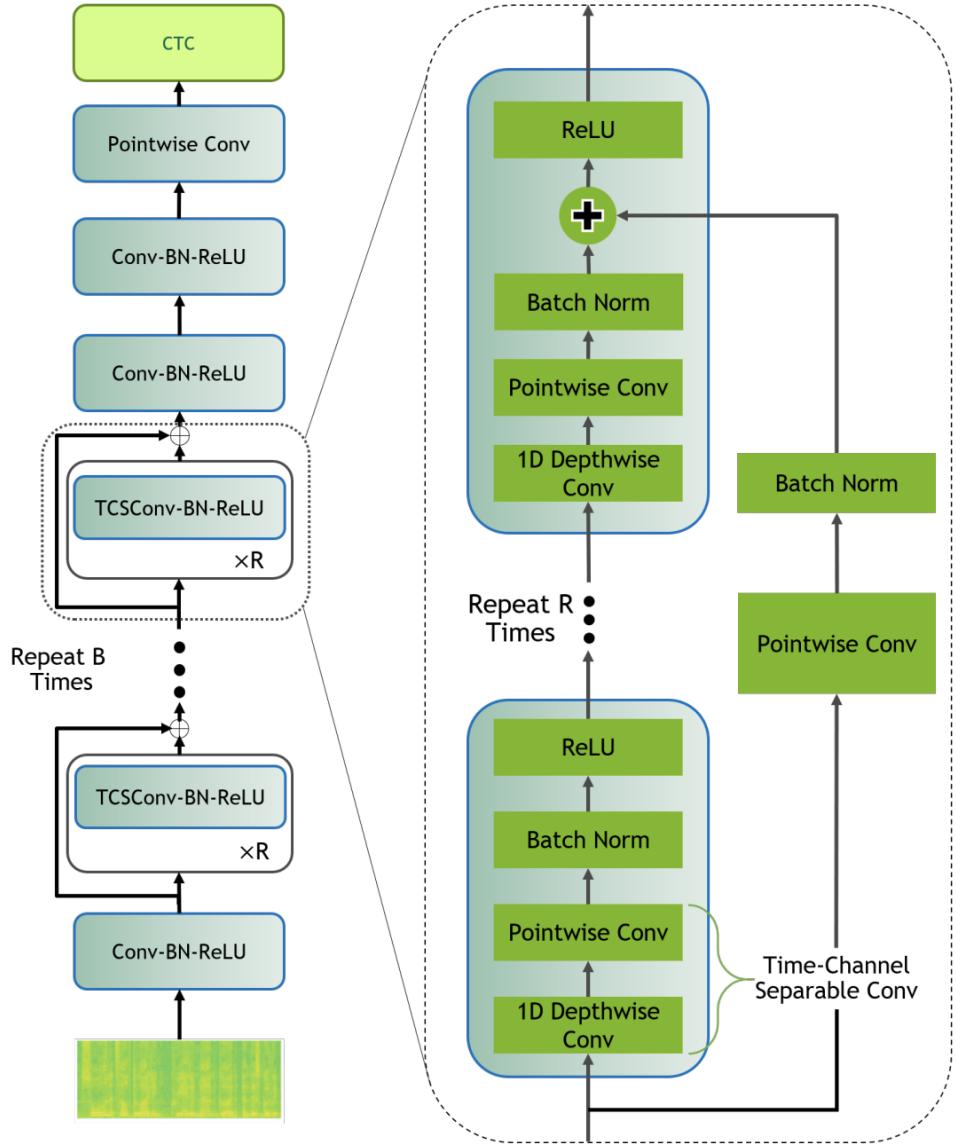


Рис. 16: Оригинальная архитектура QuartzNet 15x5

начале, так как для любого примера длина выходного текста как минимум в два раза меньше длины мэл-спектрограммы, поэтому такой трюк позволяет сократить количество вычислений вдвое. Однако, это так же уменьшает длительность каждого символа в выходе СТС. Чтобы сравнять сумму длительностей с длиной мэла, QuartzNet претерпела изменения, устанавливая `stride = 1` для первого слоя. Заметим так же что это не убирает возможность воспользоваться предобученной моделью, загрузив веса перед обучением для дообучения (fine-tuning) – размеры, форма и количество ядер (kernels) конволюций остается неизменным. Соответственно, не изменяются и размеры матриц и векторов с весами.

QuartzNet 15x5 дообучался на данных из датасета LibriTTS [17]. LibriTTS это набор данных из того же источника, что и LibriSpeech, на котором успешно обучался оригинальный QuartzNet. Однако, LibriTTS использует другую обработку дан-

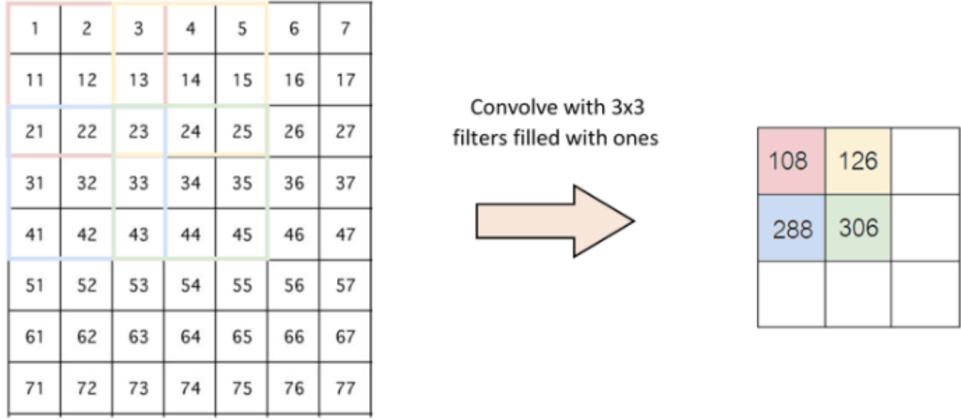


Рис. 17: Пример работы конволюций с удвоенным шагом

ных, которая более подходит для задач генерации речи, нежели распознавания речи. В частности, LibriTTS обрезает отрывки аудио по большим паузам, оставляет всю пунктуацию нетронутой (для экспрессивности речи), а также разворачиваем некоторые числа и буквенные сокращения. Для токенизации входного текста (разбиения на символы) была оставлена вся пунктуацию, давая возможность СТС самому назначить длительность каждому символу. При дообучении QuartzNet на LibriTTS достигалась побуквенная ошибка (Char-Error-Rate, CER) порядка 4.51% на части dev-clean и порядка 3.54% на тестовой части LJSpeech [15]. Выравнивание, полученное из СТС, используется для обучения предиктора длительности графемы.

Таким образом, вместо того чтобы использовать другую преодобученную TTS модель в качестве учителя для получения длительностей графем, как это делалось в модели FastSpeech [11], в рамках данной работы представлен метод в котором используется ASR модель. Ошибка, получаемая в СТС гораздо меньше ошибки при генерации речи, поэтому такой способ позволяет снять жесткое ограничение на качество, задаваемой моделью учителя.

2.2. Предсказатель длительностей графем

Первая часть TalkNet'a служит для предсказания длины мэл-спектrogramмы с помощью соответствия каждому входному символу (включая пунктуацию) количество единиц времени, требуемых для их вывода. Первым шагом предиктор длительностей вставляет пустой символ \sim между каждыми двумя соседними символами. Затем он предсказывает длительность для каждого входного символа с помощью конволюционной нейронной архитектуры. Далее, производится операция расширения (expansion) входных символов в соответствии с предсказанной длительностью (Рисунок 18).

Модель предиктора длительностей представляет собой конволюционную нейронную сеть, основанную на архитектуре модели для распознавания речи QuartzNet [24].

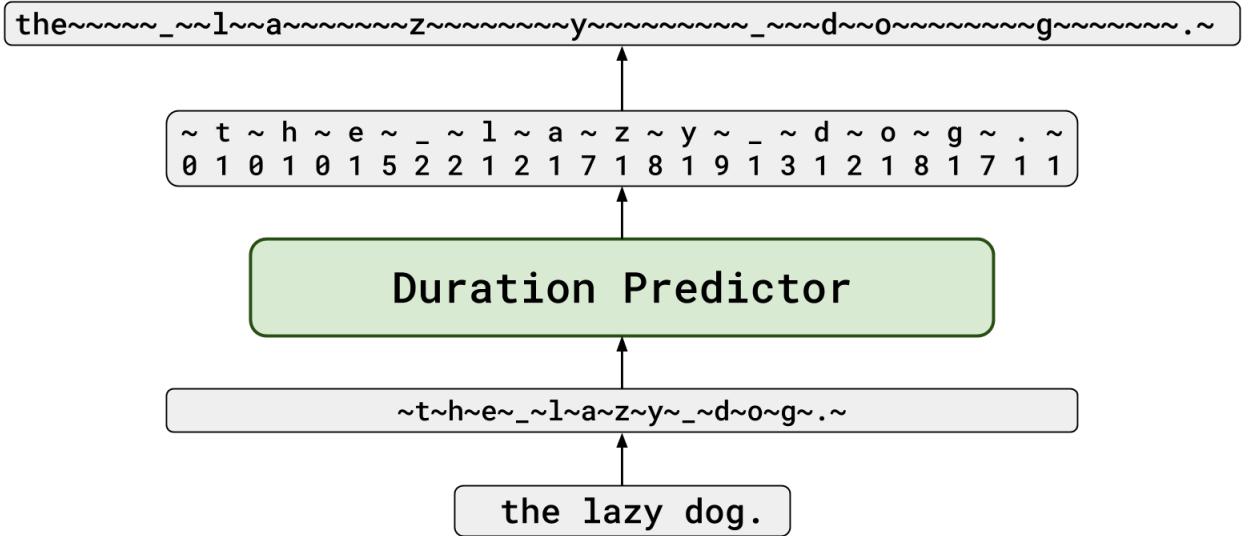


Рис. 18: Процесс предсказания длительностей графем

Модель имеет 5 больших блоков с 5 повторениями на блок. Подблок состоит из depthwise separable 15 конволюции, батч нормализации, нелинейности ReLU и дропаута (Рисунок 19). Помимо этого, применяются два дополнительных слоя: обучаемая векторизация токенов графем и 1×1 слой перед передачей в функцию потери (Таблица 1). Размерность последнего слоя зависит от типа функции потерь: для L_2 это 1, для кросс-энтропии это $|\text{множество_классов}|$.

Тренировка предиктора длительностей происходит при использовании L_2 функции ошибки с логарифмированием целевых значений аналогично [11]. Таким образом, больший вес при обучении присваивается символам с меньшими длительностями. В самом деле, разница между 15 и 16 не такая значительная как между 1 и 2. Была также испробована другая функция ошибки – кросс-энтропийные критерий, где каждый класс соответствовал определенной длительности. При классификации, были выбраны 32 самых частых класса (первые 32 длительности – от 0 до 31), а также добавлены редкие большие длительности с логарифмическим шагом после 32, так как распределение длительности графемы имеет длинный хвост (Рисунок 20). Как можно видеть, кросс-энтропия имеет несколько более высокую поклассовую точность (Таблица ??). Однако, в рамках данной работы будет использоваться L_2 , так как речь, сгенерированная с меньшим MSE для длительностей, получила несколько более высокий mean opinion score (MOS) в процессе валидации.

2.3. Генератор мэл-спектrogramм

Второй модуль производит генерацию мэл-спектrogramмы из развернутого текста. Генератор представляет собой сверточную сеть, также основанную на архитектуре

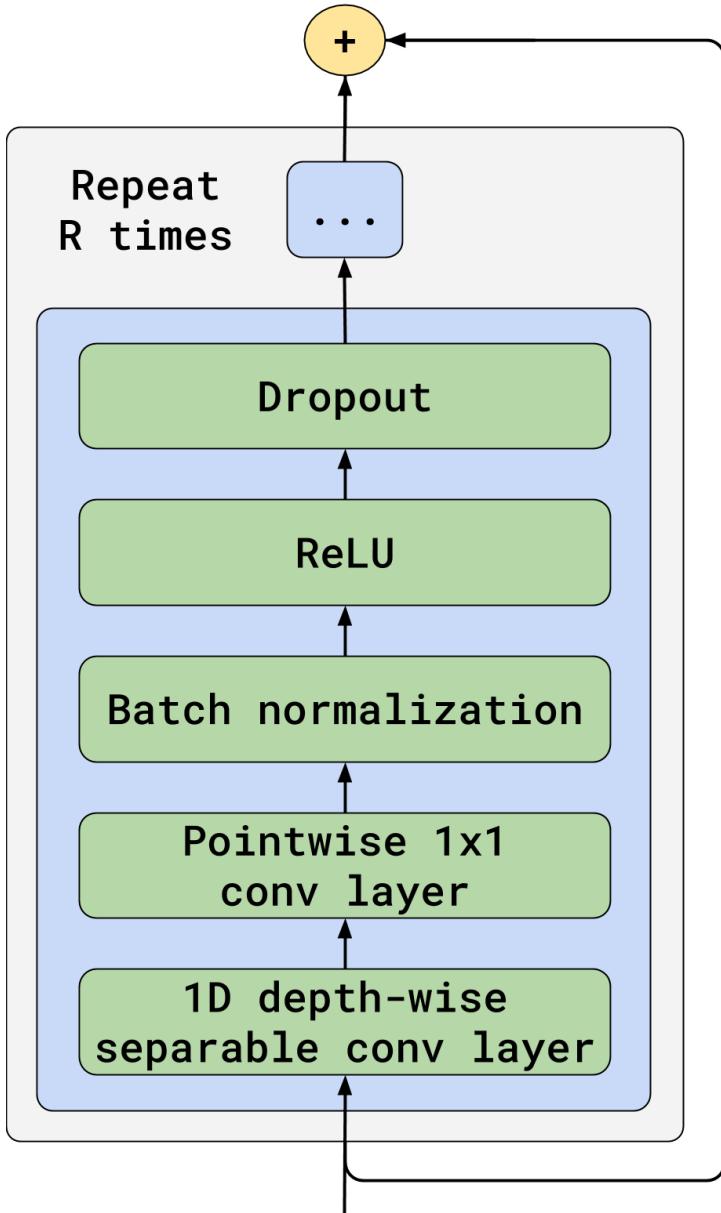


Рис. 19: Базовый блок QuartzNet. Как предиктор длительностей графем, так и генератор мэл-спектrogramм являются сверточными сетями с 1D time-channel свертками на основе QuartzNet [24].

QuartzNet. Он имеет 9 блоков с 5 подблоками (Таблица3). Как можно видеть, размер ядер конволюций увеличивается от слоя к слою с 5 до 25. Это, вкупе с residual ребрами вычислительного графа, позволяют модели выучить паттерны разного размера для применения на входной последовательности и объединить их на поздних слоях для предсказания мэл-спектrogramмы. На входе также действует слой с эмбеддингом входной последовательности и дополнительная 3x3 конволюция. На выходе стоит 1x1 конволюция с размером выхода в 80 – длиной одного мэл вектора. Генератор мэл-спектrogramм был обучен с функций ошибки – усредненной среднеквадратичной потерей (Mean-Square-Error, MSE) между элементами матриц.

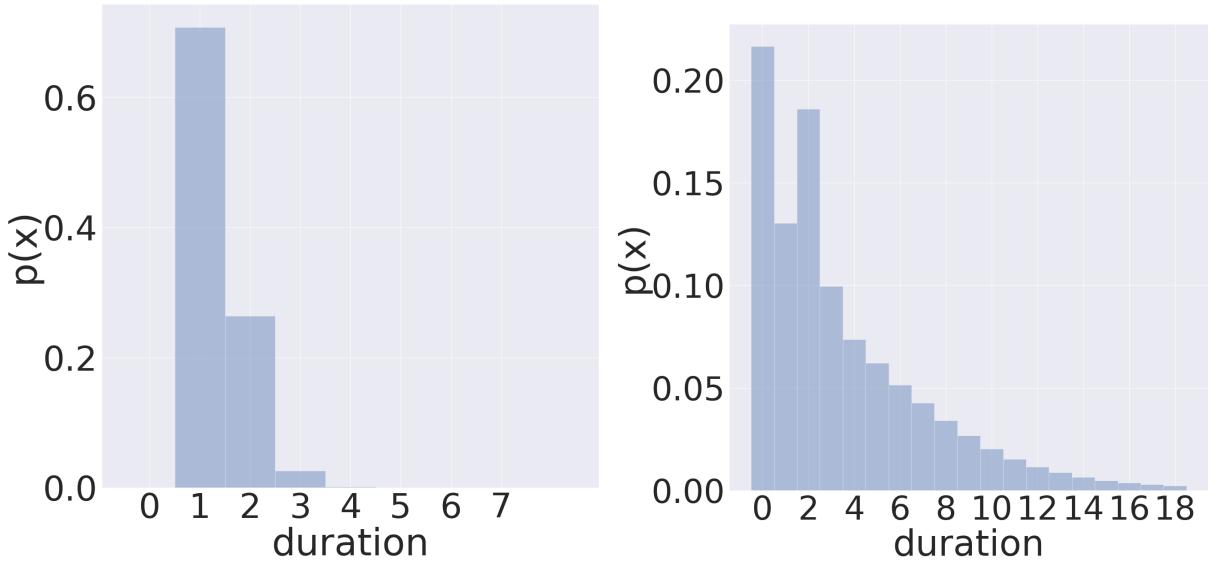


Рис. 20: Распределение длительностей исходных символов (слева) и символов перехода (справа) на основе вывода СТС для набора данных LJSpeech. Максимальная длительность для символов составляет 7, а для $\sim - 493$.

Block	# Sub Blocks	# Output Channels	Kernel Size	Dropout
Embed	1	64	1	0.0
Conv1	3	256	3	0.1
B_1	5	256	5	0.1
B_2	5	256	7	0.1
B_3	5	256	9	0.1
B_4	5	256	11	0.1
B_5	5	256	13	0.1
Conv2	1	512	1	0.1
Conv3	1	32	1	0.0
Params, M		2.3		

Таблица 1: Предиктор длительностей графем основан архитектуре QuartzNet 5x5. Residual соединения и увеличивающиеся размеры сверточ позволяют эффективно выучивать различные паттерны и комбинировать их на поздних слоях.

Пустой \sim символ дополняет алфавит исходной входной последовательности. Но вместо того, чтобы выделять для него отдельный вектор в таблице embeddings первого слоя генератора мэл-спектrogramм, он заменяется на линейную комбинацию эмбеддингов для соседним графем. Более точно, если пустой символ \sim расположен между символами a и b , его длительность равна d , то эмбеддинг E для \sim расположенного на расстоянии t слева от a будет равен $E(\sim, t) = \frac{d+1-t}{d+1} \cdot E(a) + \frac{t}{d+1} \cdot E(b)$. Это

Method	MSE	Accuracy, %	$ \mathbf{P} - \mathbf{T} \leq 1$	$ \mathbf{P} - \mathbf{T} \leq 3$
L_2	7.81	67.69	91.90	97.17
XE	10.46	69.42	92.90	97.40

Таблица 2: Результаты предиктора длительностей на тестовой части LJSpeech. P - предсказание, T - целевое значение.

более точно соответствует смысловой нагрузке пустого символа, являющегося промежуточным символом в переходе от a к b и помогает модели быстрее обучаться.

Block	# Sub Blocks	# Output Channels	Kernel Size	Dropout
Embed	1	256	1	0.0
Conv1	3	256	3	0.0
B_1	5	256	5	0.0
B_2	5	256	7	0.0
B_3	5	256	9	0.0
B_4	5	256	13	0.0
B_5	5	256	15	0.0
B_6	5	256	17	0.0
B_7	5	512	21	0.0
B_8	5	512	23	0.0
B_9	5	512	25	0.0
Conv2	1	1024	1	0.0
Conv3	1	80	1	0.0
Params, M				8.5

Таблица 3: Параметры генератора мэл-спектrogramм с архитектурой, основанной на QuartzNet 9x5

Глубина нейронной сети в 45 слоев (9 блоков по 5 подблоков) позволяет получить receptive field на последних слоях, накрывающий всю входную последовательность. Однако, он имеет неравномерную силу действия в зависимости от удаленности по времени. Это имеет простую интерпретацию: чем дальше от графемы (буквы) находится другая графема, тем меньше она влияет на правильность произношения.

В общем и целом, представленная архитектура спроектирована таким образом, чтобы максимально устранить узкие места с точки зрения скорости и эффективности реализации на графических ускорителях. Отсутствие операций с механизмом внимания, широко использующимся в других подходах, позволяет получить максимальный эффект в рамках модели многопоточности GPU. Неавторегрессионность обоих шагов,

а также явное предсказание длительностей, позволяет получить быструю устойчивую модель, сохраняя при этом возможность сохранить качество.

3. Решение

3.1. Данные для обучения

Стандартной практикой в области генерации данных является разделение датасетов на одноголосные (single-speaker) и многоголосные (multi-speaker), количество спикеров у которых доходит до нескольких тысяч, а количество минут чистой речи на каждого спикера – около 20 [17]. Предварительные эксперименты с TalkNet показали, что для обучения эффективной многоголосной системы потребуется дополнительная контекстная информация о характере голоса, без которой сложно уловить зависимости между аудио и получить хорошее качество звучания. К примеру, в качестве эмбеддинга спикера можно использовать Global Style Token [25] или похожие эксперименты. Multi-Speaker TalkNet оставлено автором как одно из направлений для будущей работы.

В рамках данной работы было решено использовать данные из известного набора LJSpeech [15], который является стандартом де-факто для тестирования процесса генерации и позволяет легко сравнивать результаты с другими подходами. LJSpeech это одноголосый набор данных из 13.100 отрывков семи аудиокниг документального жанра на английском языке. Размер отрывков варьируется от 1 до 10 секунд. Каждому отрывку в соответствие поставлена текстовая транскрипция с сохранением пунктуации и нормализацией чисел, денежных знаков и некоторых сокращений. Суммарная протяженность аудио – около 24 часов.

Набор данных был произвольно разделен на три части: 12.500 для обучения, 300 для валидации и 300 для тестирования. Для обработки текста была использована стандартная токенизация с понижением регистра и использованием пунктуации. Таким образом была полностью сохранена экспрессивность текста, что помогает модели правильным образом выучить паузы, повышение тона, выделение голосом частей текста и другие особенности речи.

Как было уже сказано выше, модель не предсказывает "сырую" аудио дорожку напрямую, а использует промежуточное представление в виде мэл-спектрограмм. Такое компактное представление строится конструктивно и детерминировано из аудио с помощью оконного преобразования Фурье. Суть этой операции в последовательном применении преобразования Фурье к коротким кусочкам речевого сигнала, доминировавшим на некоторую оконную функцию. Результат применения оконного преобразования — это матрица, где каждый столбец является спектром короткого участка исходного сигнала (Фигура 21).

Для построения мэлов использовалась библиотека `librosa`. Аудиосигналы преобразуются в мэл-спектрограммы с помощью кратковременного преобразования Фурье (Short-Time Fourier Transform, STFT), используя размер окна в 50 мс с шагом в 12.5

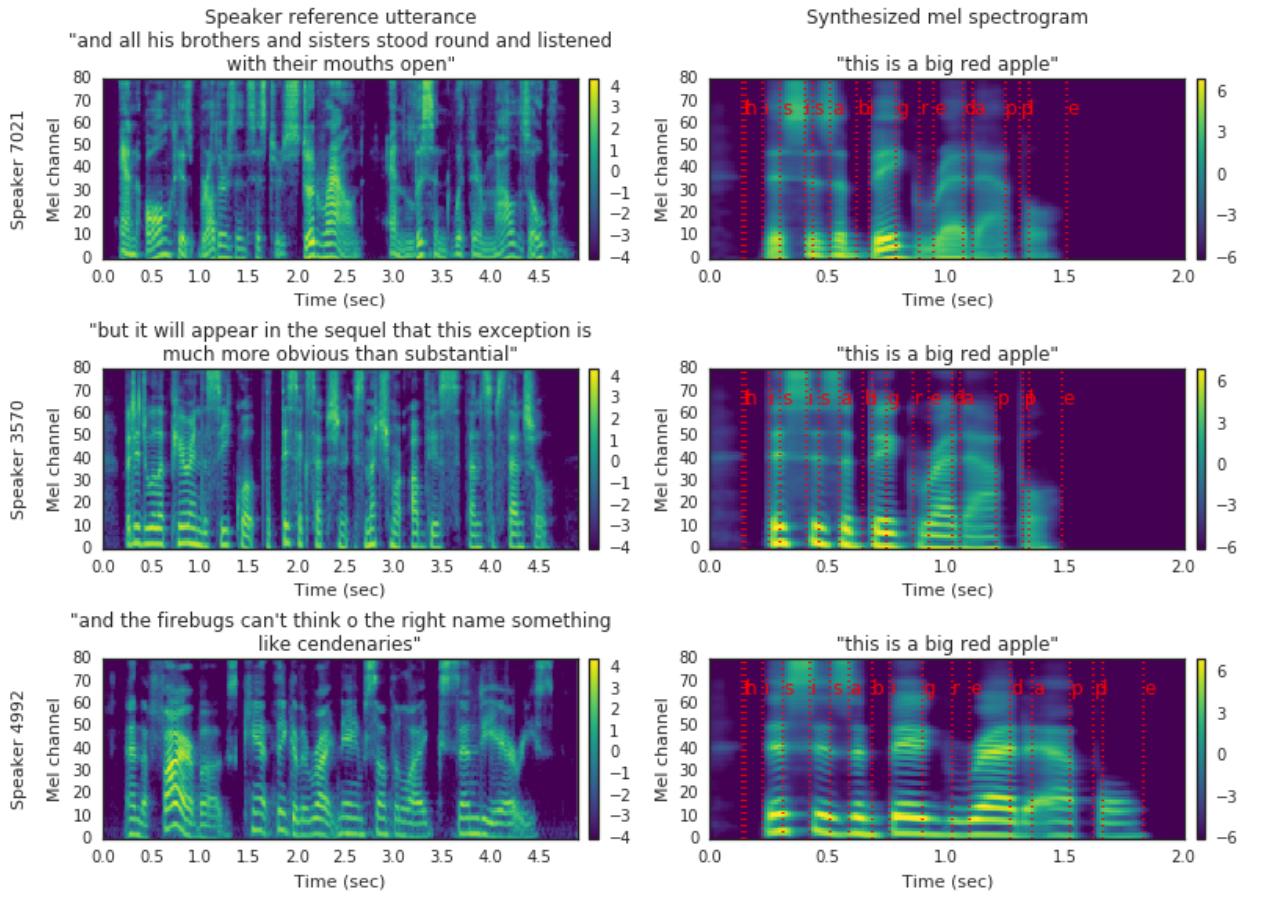


Рис. 21: Примеры полученных мэл-спектрограмм с выровненным текстом

мс, окно вида "hann" и логарифмируем результат. Более подробные характеристики преобразования: `win_length = 1024, hop_length = 256, n_fft = 1024, low_freq = 0` и `high_freq = 80`.

3.2. Обучение предсказателя длительностей графем

Как уже было сказано выше, часть TalkNet'a ответственная за предсказание длительности графем обучается отдельно. На вход такой модели подается текстовое представление токенизированное посимвольно. Далее, между каждыми соседними символами вставляется пустой (blank, ~) символ, означающий промежуточное состояние для перевода дыхания, кратковременной паузы и итд (Рисунок 18). Итого, общая длина входа удваивается, как и длина выхода. Сама же модель предсказания основана на нейронной неавторегрессионной конволюционной архитектуре QuartzNet 5x5.

Нейронная модель для предсказания длительности графемы обучалась с помощью оптимизатора Adam с $\beta_1 = 0.9, \beta = 0.999, \epsilon = 10^{-8}$, `weight_decay = 10-6` и `gradient_norm_clipping = 1.0`. Для `learning_rate` использовалась нелинейная зависимость cosine decay policy начиная от 10^{-3} до 10^{-5} с `warmup = 0.02`. Для обучения

ния использовались различные конфигурации вычислительных мощностей с 1 и 8 графическими процессорами (GPU) V100 с 16 и 32 гигабайтами видеопамяти. Тренировка проводилась с батчем размера 256 для одной GPU в 16GB и увеличивали `learning_rate` пропорционально увеличению мощностей. Такая конфигурация гиперпараметров позволила получить сходимость всего лишь за 200 эпох, что занимало около 1.3 часа чистого времени на одной GPU и около 11 минут на восьми GPU. Также, было использовано обучение со смешанной точностью (mixed-precision, [18]), так как эмпирически было выявлено что такой подход позволяет получить почти двое-кратное ускорение с сохранением точности.

Результаты можно увидеть на Таблице 2. Как можно заметить, нам удалось получить почти 70% точности с простой моделью, которая содержит около 2.3 миллиона весов. Более того, около 92% предсказаний находятся на абсолютном расстоянии не более чем в 1.

3.3. Обучение генератора мэл-спектrogramм

Генератор мэл-спектrogramм производится из символьной входной последовательности после операции расширения (expansion), в результате которой длина текста и длина мэла выравнивались по времени (Рисунок 3). Такое выравнивание соотносит буквы с произносимыми звуками и переходами между ними, облегчая процесс генерации. В качестве истинных длительностей графем для тренировки используются длительности полученные на этапе извлечения. Таким образом, обучение генератора не требует использования предобученного предиктора длительностей и может выполняться параллельно. Сама же модель генератора основана на нейронной неавторегрессионной конволюционной архитектуре QuartzNet 9x5.

Как видно из Таблицы 2, точность предсказателя длительностей составляет около 70%. В то же время, количество классов находящихся на абсолютном расстоянии не более 1 - около 92%. Для того чтобы уменьшить несоответствие, которое возникает на этапе вывода (inference), были применены аугментации для истинных длительностей подающихся к обоим частям модели. Такие аугментации должны отвечать некоторых заданным критериям:

1. Сохранять сумму длительностей всех букв неизменной, так как она напрямую зависит от длины мэл-спектrogramмы.
2. Быть несмещеными относительно истинных длительностей.
3. Сила изменений для каждого символа должна быть пропорциональна длительности. Таким образом, символы с большими длительностями будут изменяться чаще.
4. Сила аугментации должна быть контролируема с заданным параметром.

Одной из аугментаций, удовлетворяющих всем вышеперечисленным условиям, может являться "биномиальная встряска" (binomial shake). Суть заключается в "обмене" длительностями соседних символов l и r по биномиальному распределению с заданным параметром p и $n = \min(d_l, d_r)$. Каждый символа обменивается длительностями с двумя соседями, а направление обмена выбирается случайным образом с вероятностью $p = 0.5$ (Рисунок 22).

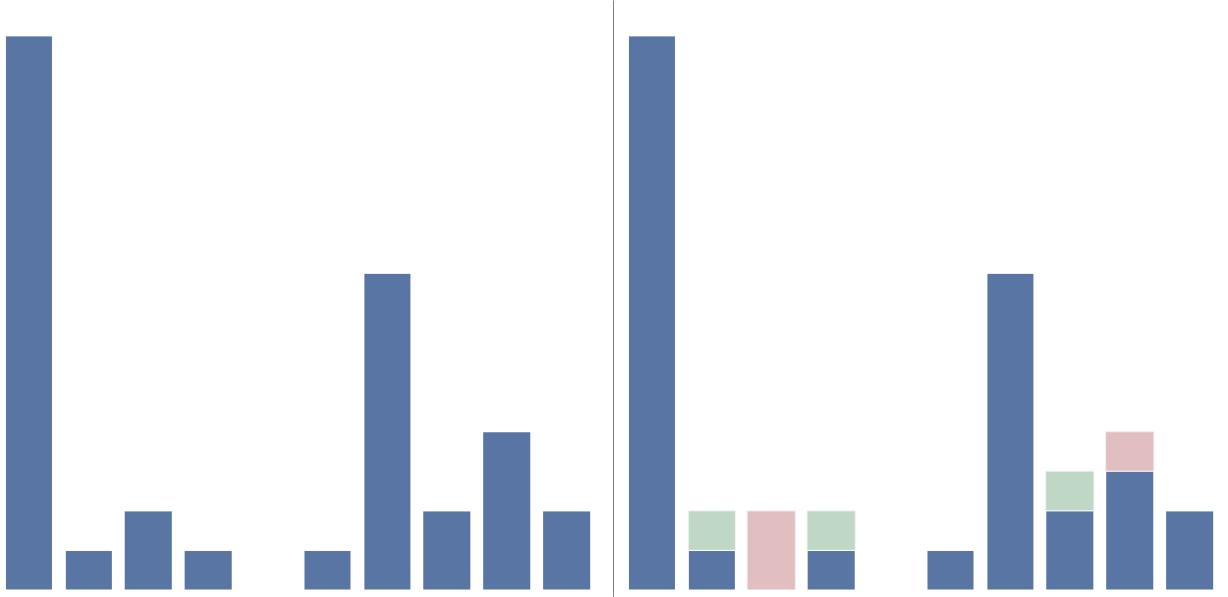


Рис. 22: Пример применения аугментации для длительностей графем для соседних символов. Слева - до, справа - после.

Опытным путем было выяснено, что аугментации помогают качеству звучания и уменьшают эффект переобучения (overfitting). Для тренировки генератора мэлспектрограмм применялась "биномиальную встряску" с $p = 0.05$ для аугментации истинных длительностей.

Для тренировки генератора мэлов использовался тот же набор гиперпараметров, что и для предсказателя графем. Использовался батч размера 64 для одной GPU в 16GB и увеличивали `learning_rate` пропорционально увеличению мощностей. Такая конфигурация позволила получить сходимость всего лишь за 200 эпох, что занимало около 8 часов чистого времени на одной GPU и около 2 часов на восьми GPU. Также, было использовано обучение со смешанной точностью (mixed-precision, [18]), так как эмпирически было выявлено что такой подход позволяет получить почти двоекратное ускорение с сохранением качества.

Таким образом, процесс обучения обеих частей TalkNet'a на сервере DGX-1 может занимать всего порядка 2-ух часов. Это сравнимо меньше 2 – 3 дней которые требуются модели Tacotron2 [19]. Такая особенность связана прежде всего с эффектом неавторегрессионности, а также отсутствием операций основанных на механизмах внимания (attention), которые обычно занимают порядка $O(T^2)$ времени в зависимости

ости от длины T .

4. Результаты

4.1. Качество аудио

Одна из самых больших проблем с разработкой систем для генерации речи это отсутствие быстро (программно) вычислимой метрики, которая хорошо коррелирует с качеством произношения. Зачастую, в процессе разработки, качество приходиться мерить "на слух", замедляя таким образом исследования и ухудшая реальную требуемую корреляцию с человекоподобной речью. Для того, чтобы оценить качество TalkNet, следуя другим работам [11, 19], было решено провести случайный слепые тесты с дискретным оцениванием и усреднить результаты для того, чтобы получить примерное представление о работоспособности получившейся модели.

Для оценки качества был проведен эксперимент MOS (mean opinion score) для сгенерированной речи с использованием Amazon Mechanical Turk [1]. Сравнивались четыре набора подходов к генерации для тестовой части датасета LJSpeech: 1) Истинные аудио с речью; 2) Истинные мэл-спектrogramмы, преобразованная в речь с помощью WaveGlow; 3) Tacotron 2 + WaveGlow и 4) TalkNet + WaveGlow. Были использованы реализации NVIDIA для Tacotron 2 и WaveGlow. Тестились 100 аудио примеров, где каждый пример был оценен не менее 10 раз 10 различными людьми. Также, были использованы дополнительные фильтры для отсева людей без высшего образования или людей не знающих английский язык для повышения стабильности оценивания. Оценивающим предлагалось проверить работоспособность гарнитуры, несколько раз прослушать отрывок и выбрать наиболее подходящую оценку на вопрос "насколько представленный отрывок похож на человеческую речь?". Баллы варьировались от 1.0 до 5.0 с шагом 0.5 (всего – 9 ступеней). Как можно заметить, качество речи TalkNet сравнимо к Tacotron 2 (Таблицы 4).

Model	MOS
Ground truth speech	4.31 ± 0.05
Ground truth mel + WaveGlow	4.04 ± 0.05
Tacotron 2 + WaveGlow	3.85 ± 0.06
TalkNet + WaveGlow	3.74 ± 0.07

Таблица 4: Усредненные оценки MOS (mean opinion score) с 95% доверительным интервалом

Одна из дополнительных характеристик генерации, помимо качества звучания, это способность модели не пропускать слова и правильно произносить имеющиеся даже при сложных входных строках. Такая характеристика называется устойчивостью

(robustness). Следуя FastSpeech [11], были проведены эксперименты для сравнения устойчивости различных моделей на 50 особенно сложных для генерации входных предложениях 23, считая вручную количество пропущенных или повторенных слов. Данные предложения представляют особенную сложность, из-за отсутствия правдоподобного контекста, поэтому генеративной модели приходится выводить корректное произношение почти побуквенно. В результаты экспериментов было обнаружено, что подобно FastSpeech, TalkNet устраниет ошибки связанные с пропущенными или повторяющимися словами. Таким образом, неавторегрессионный подход очень устойчив и не имеет ошибок с отсутствующими или повторяющимися словам, что делает его более выгодным по сравнению с авторегрессионными моделями TTS, такими как Tacotron 2 или Transformer TTS.

01. a
02. b
03. c
04. H
05. I
06. J
07. K
08. L
09. 2222222 hello 2222222
10. S D S D Pass zero - zero Fail - zero to zero - zero - zero Cancelled - fifty nine to three - two - sixty four Total - fifty nine to three - two -
11. S D S D Pass - zero - zero - zero Fail - zero - zero - zero - zero Cancelled - four hundred and sixteen - seventy six -
12. zero - one - one - two Cancelled - zero - zero - zero Total - two hundred and eighty six - nineteen - seven -
13. forty one to five three hundred and eleven Fail - one - one to zero two Cancelled - zero - zero to zero zero Total -
14. zero zero one , MS03 - zero twenty five , MS03 - zero thirty two , MS03 - zero thirty nine ,
15. 1b204928 zero one seven ole32

Рис. 23: Первые 15 из 50 сложных примеров для проверки генерации TTS систем

4.2. Скорость генерации

Процесс вывода (inference) TalkNet описан на Рисунке 3. Сначала, вставляются пустые символы в входной текст между каждыми двумя соседними. Полученная последовательность пропускается через предиктор длительностей графем. Входные данные предиктора длительностей затем корректируются для символов с длительностью 0. Так избегаются неправильные предсказания длительностей для редких символов (знаков препинания), что позволяет оставить их после операции расширения. Исправленная последовательность символов расширяется при повторении каждого символа

в соответствии с предсказанной длительностью. Вторая часть модели генерирует мэл-спектrogramму из развернутой последовательности графем, равной ей по длине.

Такой процесс имеет несколько узких мест с точки зрения скорости:

- Матричные операции, необходимые для вывода предиктора длительностей.
- Матричные операции, необходимые для вывода генератора мэл-спектrogramм.
- Передача и трансформация данных между частями.

Очевидно, что все части вывода по времени зависят от длины текста и соответствующего мэла. Поэтому, для правильного сравнивания TTS систем для генерации речи замерялась задержка (*latency*), необходимая всем частям системы суммарно на получение мэл-спектrogramмы. Получившиеся задержки усреднялись ее по множеству примеров с различной длиной. Заметим также, что этап вокодинга не включен во время задержки, но все еще необходим для построения полной системы для генерации речи. Обычно, вокодинг занимает в десятки раз больше времени и весов, поэтому все еще является узким местом всей системы.

Были сравнены задержки вывода TalkNet с Tacotron 2 и FastSpeech. Использовались реализации FastSpeech от NVIDIA, так как исходный код был недоступен на момент оценки. Чтобы измерить задержку, были сгенерированы мэл-спектrogramмы с размером батча, равным 1 и проводим усреднение по 2048 образцам из тестового набора данных LJSpeech. Средняя длина мэл-спектrogramмы при таком подходе составляет 520. Сравнивались задержки используя один и то же аппаратное оборудование (*hardware*) – один графический процессор V100. Как можно видеть, скорость вывода TalkNet значительно быстрее, чем Tacotron 2 и FastSpeech (Таблица 5). Если же проводить вывод по 4 или 8 примеров за раз, то скорость повышается линейно, достигая 1300 RTF.

Поскольку TalkNet не использует операции с механизмами внимания (*attention*), задержка вывода практически не зависит от длины входного сигнала 24. Операции с *attention* обычно реализуются с перемножением матриц, вытянутых по длине вывода. Поэтому, такая операция занимает порядка $O(T^2)$ операция, что приводит к линейному росту времени при параллелизации на графических ускорителях ГПУ.

Стоит также отметить, что текущим самым узким местом с точки зрения скорости вывода TalkNet является непосредственные вычисления операций нейронной сети (конволюции, нелинейности, батч нормализация, дропаут). Однако, текущие реализации CUDA кернелов (специальных функций для вычислений операций на графических ускорителях) для depthwise separable конволюций, из которых почти полностью состоит TalkNet, написаны не самым оптимальным образом. Полагается, что написание оптимальных функций для 1d time и 1x1 pointwise сверток с использованием

Model	# Batch size	Inference Latency, s	RTF
Transformer TTS [21]	1	6.735 ± 3.969	1.48 ± 0.87
Tacotron 2 [19]	1	$0.817 \pm 1 \cdot 10^{-2}$	7.56 ± 0.01
FastSpeech [11]	1	$0.029 \pm 2 \cdot 10^{-4}$	221.01 ± 1.75
TalkNet	1	$0.019 \pm 1 \cdot 10^{-5}$	328.65 ± 4.76
TalkNet	4	$0.023 \pm 5 \cdot 10^{-5}$	1048.80 ± 21.75
TalkNet	8	$0.037 \pm 4 \cdot 10^{-4}$	1340.09 ± 8.90

Таблица 5: Задержка вывода TalkNet для генерации мэл-спектрограммы (без вокодера). Задержка была измерена с размером батча 1 с использованием графического процессора V100 и усреднена по 2048 примерам из набора данных LJSpeech. Приведены усредненное время задержки и фактор реального времени (RTF) с доверительным интервалом 95%. Фактор реального времени показывает сколько секунд речи можно сгенерировать за одну секунду вычислений.

идей fusing’а (когда две последовательные операции сокращаются в одну) может значительно ускорить работу предложенной модели.

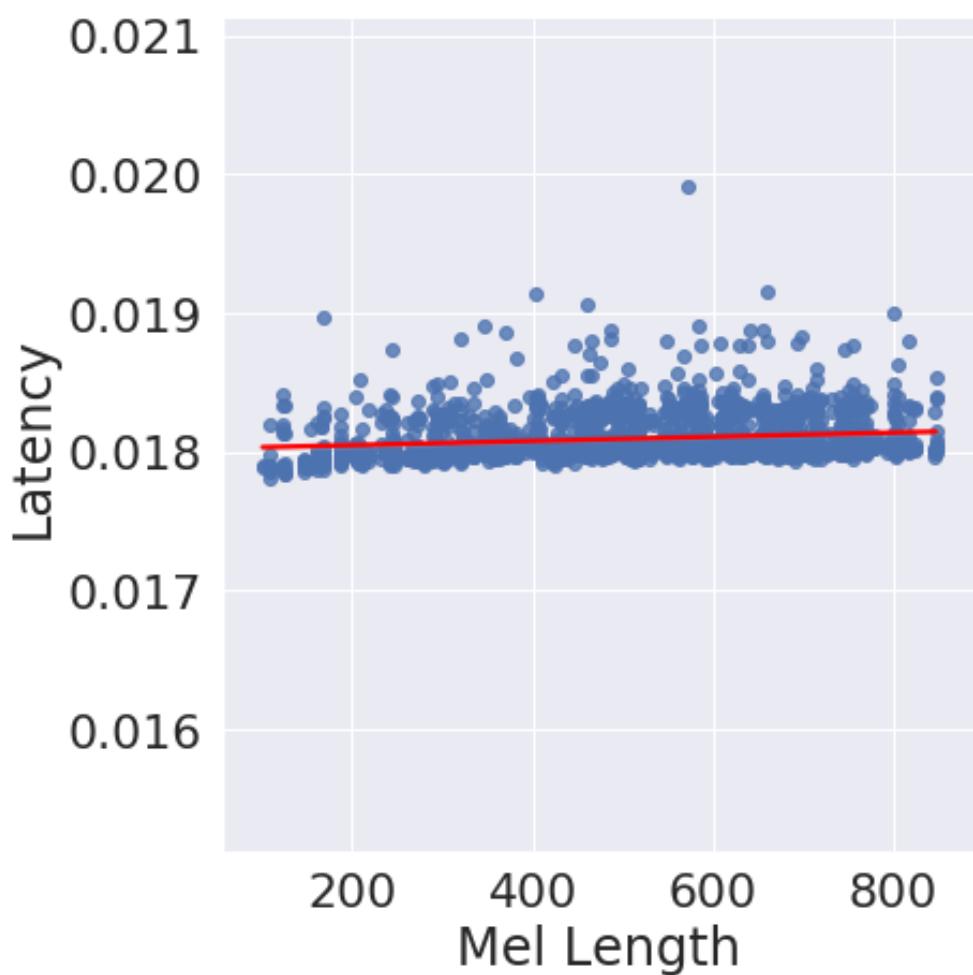


Рис. 24: Влияние неавторегрессионной архитектуры и отсутствия механизмов внимания

Заключение

В рамках данной работы был описан TalkNet [4], нейронная система синтеза речи основанная по конволюционных сетях. Модель состоит из двух сверточных сетей: предиктора длительностей графем и генератора мэл-спектrogramм. Эта модель не требует другой предобученной системы в качестве учителя. Истинное выравнивание графем извлекается из выходных данных СТС для предварительно обученной модели распознавания речи.

Предсказывание длительностей явным образом практически исключает возможность пропущенных или повторяющихся слов. TalkNet достигает сопоставимого уровня качества речи с Tacotron 2 и FastSpeech. Данный подход также обладает свойством компактности. Он имеет только 10.8 миллионов обучаемых параметров, что почти в 3 раза меньше, чем предлагают аналогичные нейронные модели: Tacotron 2 имеет 28.2 миллиона, а FastSpeech имеет 30.1 миллиона параметров. Обучение TalkNet занимает всего около 2 часов на сервере с 8 графическими ускорителями V100. Параллельная генерация мэл-спектrogramмы делает скорость обучения и вывода значительно быстрее конкурентов.

Современные генеративные модели все еще обладают рядом фундаментальных проблем, которые не позволяют считать задачу генерации речи решенной. Ценность этой работы заключается не только в предложенном и описанном подходе, позволяющем сильно сократить время и количество параметров, требуемых TTS системе, но и в трудностях, возникших при реализации, тестировании и сравнении подходов, указывающих на глобальные проблемы и очерчивающих границы применимости того или иного метода или модели.

В рамках данной работы можно выделить следующие основные результаты:

- Проанализирована предметная область и существующие модели. Обозначены основные проблемы и намечены пути к их решению.
- Разработана новая неавторегрессионная конволюционная архитектура, не требующая предобученной TTS системы в качестве учителя.
- Произведены сравнение подходов и анализ результатов на качество и скорость. Предложенный подход показал сравнимое качество, являясь при этом быстрее и компактнее аналогичных методов.

Представленный подход, несмотря на свою простоту, открывает сразу несколько направлений для дальнейшего изучения и доработки.

- Основное направление - улучшение качества. Следуя аналогичным статьям, переход с графем на звуковые фонемы должен помочь модели правильнее произносить звуки в словах.

- TalkNet multi-speaker. Многоголосовое расширение позволит проще искать данные для обучения и увеличит число потенциальных применений в индустрии.

Модель, скрипт для обучения и сгенерированные примеры будут выложены с открытым исходным кодом как часть библиотеки NeMo [20]. Автор благодарят Джона Коэна, Виталия Лаврухина, Джейсона Ли, Кристофера Паризьена и Жоао Фелипе Сантоса за полезные отзывы и рецензии.

Список литературы

- [1] Amazon. — Amazon Mechanical Turk, 2005. — Access mode: <https://www.mturk.com/>.
- [2] Attention is all you need / Ashish Vaswani, Noam Shazeer, Niki Parmar et al. // NIPS. — 2017.
- [3] Bahdanau Dzmitry, Cho Kyunghyun, Bengio Yoshua. Neural machine translation by jointly learning to align and translate // arXiv:1409.0473. — 2014.
- [4] Beliaev Stanislav, Rebryk Yurii, Ginsburg Boris. TalkNet: Fully-Convolutional Non-Autoregressive Speech Synthesis Model. — 2020. — 2005.05514.
- [5] Biopython: freely available Python tools for computational molecular biology and bioinformatics / Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang et al. // Bioinformatics. — 2009. — 03. — Vol. 25, no. 11. — P. 1422–1423. — <https://academic.oup.com/bioinformatics/article-pdf/25/11/1422/944180/btp163.pdf>.
- [6] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep Residual Learning for Image Recognition. — 2015. — 1512.03385.
- [7] Deep Voice 2: Multi-Speaker Neural Text-to-Speech / Andrew Gibiansky, Sercan Arik, Gregory Diamos et al. // NIPS. — 2017.
- [8] Deep Voice 3: 2000-Speaker Neural Text-to-Speech / Wei Ping, Kainan Peng, Andrew Gibiansky et al. // ICLR. — 2018.
- [9] Deep Voice: Real-time Neural Text-to-Speech / Sercan Ömer Arik, Mike Chrzanowski, Adam Coates et al. // arXiv:1702.07825. — 2017.
- [10] Fast, Compact, and High Quality LSTM-RNN Based Statistical Parametric Speech Synthesizers for Mobile Devices / Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts et al. // INTERSPEECH. — 2016.
- [11] FastSpeech: Fast, Robust and Controllable Text to Speech / Yi Ren, Yangjun Ruan, Xu Tan et al. // NeurIPS. — 2019.
- [12] Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi et al. Generative Adversarial Networks. — 2014. — 1406.2661.
- [13] Hannun Awni. Sequence Modeling with CTC // Distill. — 2017. — <https://distill.pub/2017/ctc>.

- [14] Heiga Zena Keiichi Tokudaa Alan W. Blackc. — Statistical Parametric Speech Synthesis, 2009. — Access mode: https://www.cs.cmu.edu/~pmuthuku/mlsp_page/lectures/spss_specom.pdf.
- [15] Ito Keith et al. The LJ speech dataset. — 2017.
- [16] Kaiser Lukasz, Gomez Aidan, Chollet Francois. Depthwise separable convolutions for neural machine translation // arXiv:1706.03059. — 2017.
- [17] LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech / Heiga Zen, Rob Clark, Ron J. Weiss et al. // INTERSPEECH. — 2019. — Access mode: <https://arxiv.org/abs/1904.02882>.
- [18] Mixed precision training / Paulius Micikevicius, Sharan Narang, Jonah Alben et al. // arXiv:1710.03740. — 2017.
- [19] Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions / J. Shen, R. Pang, R. J. Weiss et al. // ICASSP. — 2018.
- [20] NeMo: a toolkit for building AI applications using Neural Modules / Oleksii Kuchaiev, Jason Li, Huyen Nguyen et al. // arXiv:1909.09577. — 2019.
- [21] Neural Speech Synthesis with Transformer Network / Naihan Li, Shujie Liu, Yanqing Liu et al. // AAAI. — 2019.
- [22] Parallel Neural Text-to-Speech / Kainan Peng, Wei Ping, Zhao Song, Kexin Zhao // arXiv:1905.08459. — 2019.
- [23] Prenger R., Valle R., Catanzaro B. WaveGlow: A Flow-based Generative Network for Speech Synthesis // ICASSP. — 2019.
- [24] QuartzNet: deep automatic speech recognition with 1D time-channel separable convolutions / Samuel Kriman, Stanislav Beliaev, Boris Ginsburg et al. // ICASSP. — 2020.
- [25] Wang Yuxuan, Stanton Daisy, Zhang Yu et al. Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis. — 2018. — 1803.09017.
- [26] Tacotron: Towards End-to-End Speech Synthesis / Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton et al. // INTERSPEECH. — 2017.
- [27] Taylor Paul. Text-to-Speech Synthesis. — Cambridge University Press, 2009.
- [28] VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop / Yaniv Taigman, Lior Wolf, Adam Polyak, Eliya Nachmani // arXiv:1707.06588. — 2017.

- [29] Zen H., Sak H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis // ICASSP. — 2015.