Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский
Академический университет Российской академии наук»
Центр высшего образования

Кафедра математических и информационных технологий

Курбанов Рауф Эльшад оглы

Генерация речи с учётом индивидуальных особенностей

Магистерская диссертация

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

Научный руководитель:
Шпильман А. А.

Рецензент:
Тузова Е. А.

Санкт-Петербург
2017

SAINT-PETERSBURG ACADEMIC UNIVERSITY
Higher education centre

Department of Mathematics and Information Technology

Rauf Kurbanov

Speech generation with individual characteristics

Graduation Thesis

Admitted for defence.
Head of the chair:
professor Alexander Omelchenko

Scientific supervisor:
Alexey Shpilman

Reviewer:
Ekaterina Tuzova

Saint-Petersburg
2017

Оглавление

Введение **5**

1. Обзор решений **8**
1.1. Генеративные модели 8
1.2. Генерация речи на основе скрытых марковских моделей 8
1.2.1. Преимущества генерации на основе CMM 9
1.3. Генерация речи на основе глубинного обучения 9
1.3.1. Глубинные архитектуры для генерации речи 10
1.3.2. Deep Speech 10
1.4. Сравнение с существующим решением 11

2. WaveNet **12**
2.1. Свёртки 13
2.2. Дырявые свёртки 14
2.3. Gated activation unit 15
2.4. Residual и skip соединения 16
2.5. WaveNet с условием 17
2.5.1. Локальные и глобальные условия 17

3. Основные цели **18**

4. Методы и реализация **19**
4.1. Описание данных 19
4.2. Реализация модели 19
4.2.1. Обработка условий 23
4.2.2. Глобальные условия 23
4.2.3. Локальные условия 23
4.3. Извлечение признаков 25
4.3.1. Признаки для представления данных 26
4.3.2. Признаки для глобального условия 26
4.3.3. Признаки для локального условия 27
4.3.4. Выравнивание аудио 27

5. Обсуждение результатов **29**
5.1. Сравнение качества результатов 29
5.2. Производительность 31
5.3. Анализ экспериментов 31

6. Заключение **33**
6.1. Направления развития 33

Список литературы **34**

Введение

В данной работе исследуется метод генерации голоса с дополнительными характеристиками. Главным вдохновением к проведённой работе послужили недавние продвижения в области нейросетевых генеративных архитектур, способных генерировать такие сложные вероятностные распределения как человеческая речь[13].

Мы проведём обзор самых современных подходов к генерации голоса и сравним существующие решения с моделью на основе архитектуры WaveNet 1.

Далее исследуем нюансы реализации легковесной модели для генерации голоса, показывающей state of the art качество голоса 2. А также поясним теоретическую базу модификации такой архитектуры для условной генерации 2.5.

Центаральная часть работы посвящена деталям реализации упомянутой архитектуры а также проектированию и реализации признаков, передаваемых в качестве условий 4.1.

В заключении мы проведём эксперименты с разными конфигурациями модели и проанализирем результаты 5.

Мотивация

В сообществе специалистов в области машинного обучения последние годы прослеживается пока ещё незатухающий интерес к нейронным сетям. На многих публичных выступлениях любят говорит о том, что дешёвые GPU дали второе дыхание нейронным сетям. Вы часто услышите об открытых соревнованиях, на которых решения, основанные на глубинном обучении, добились такой точности, что отправили считавшуюся ранее открытой проблему в разряд решённых. Одним из таких знаменитых примеров стало закрытие проекта Asirra[1] в результате контекста на Kaggle[6].

В этой работе мы не преследуем целей добиться выдающихся результатов с точки зрения точности модели. Дело в том, что самые последние исследования section 1 в сфере генерации голоса публикуются исследовательскими командами крупных корпорации с большими вычислительными ресурсами.

Статья [13], на результатах которой основана данная работа не исключение. Мы не собирались соревноваться в качестве генерации с существующими решениями. Мы, в свою очередь, постараемся развить идею придачи дополнительных характеристик генерируемому голосу. Такие характеристики можно придумать самые разные: от простого мужской/женский голос до имитации речи конкретного человека.

Инструментарий для таких целей предоставляет публикация WaveNet: A Generative Model for Raw Audio [13], однако только с точки зрения архитектуры и совсем без описания дополнительных признаков. Мы посчитали, что более подробное исследование этого вопроса и получение практических результатов было бы достойным вкладом в исследовательское сообщество.

Практическая ценность

Модель, которую мы хотим построить в перспективе может найти множество практических применений от персонализированного голоса в навигаторе до эмуляции голосов знаменитостей. Конечно разработка такой полноценной промышленной системы самостоятельная сложная задача, выходящая далеко за рамки данной работы. Однако мы должны понимать, как должен выглядеть внешний интерфейс взаимодействия с моделью, чтобы последняя была полезна на практике.

Высокоуровневый макет предполагаемой системы изображен на рисунке 1.

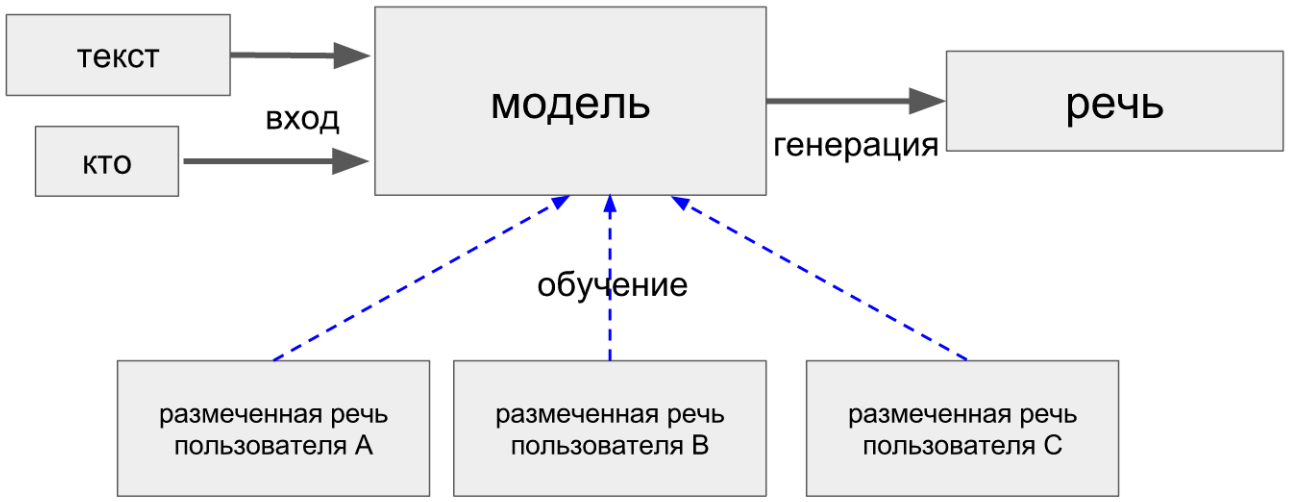

Рис. 1: Система генерации речи

Опишем, каким бы мы хотели видеть сценарий использование системы, в предположении что подлежащая модель работает идеально. Изображенная система должна быть заранее обучена на некоторой базе пользователей, для каждого из которых предоставлено достаточное количество записей голоса вместе с выровненным вдоль звука произносимым текстом.

На вход должен подаваться текст, вместе с дополнительными признаками, которые мы назовём **локальные условия** и некоторой общей харакеристкой, к примеру, описывающей, чей голос мы хотим сгенерировать. Такую характеристику назовём **глобальным условием**.

На выходе возвращается речь с ожидаемыми особенностями.

1. Обзор решений

Методология генерации голоса имеет достаточно богатую историю на протяжении большей части которой в основе самых передовых на тот момент методов лежали системы на основе скрытых марковских моделей. Во многом благодаря тому, что стоит задача построения генеративной модели, марковские свойства до сих присутствуют, однако могут ослабляться, как, например, в нашем решении. Современные системы генерации голоса всё больше основываются на глубинном обучении как в качестве вспомогательного механизма, так и полной замены скрытых марковских моделей.

1.1. Генеративные модели

В статистике и машинном обучении генеративная модели это модель для генерации данных при наличие скрытых параметров. Модель определяет совместное распределение вероятности над пространством наблюдений. Генеративные модели используются в машинном обучении либо для генерации данных напрямую, либо в качестве промежуточного шага для получения плотности распределения функции условной вероятности. Условное распределение может быть получено из генеративной модели с помощью теоремы Байеса.

Генеративные модели противопоставляются дискриминативным моделям в том плане, что генеративная описывает полную вероятностную модель для всех переменных, когда дискриминативная в свою очередь моделирует лишь вероятностное распределение лишь целевой переменной. Таким образом генеративная модель может быть использована для симуляции значений любой переменной, используемой в модели, когда дискриминативная дает возможность лишь получить целевую переменную по наблюдаемым признакам. Несмотря на то, что дискриминативным моделям не требуется симулировать распределение наблюдаемой переменной, они зачастую не могут описать более сложные отношения между признаками и целевой переменной. Такие модели не обязательно показывают лучше результаты в задачах классификации и регрессии. В современных приложениях данные два класса моделей зачастую служат разными подходами к одной и той же процедуре. [14]

1.2. Генерация речи на основе скрытых марковских моделей

Определение 1.1 ***Скрытая марковская модель (СММ)*** — *статистическая модель, имитирующая работу процесса, похожего на марковский процесс с неизвестными параметрами, и задачей ставится разгадывание неизвестных параметров на основе наблюдаемых. Полученные параметры могут быть использованы в дальнейшем анализе, например, для распознавания образов. СММ может быть рассмотрена как простейшая байесовская сеть доверия. [15]*

- Поменяв местами вход и выход скрытой марковской модели в задаче распознавания голоса, мы обращаем скрытую марковскую модель в генеративную модель для задачи трансляции текста в звук.
- Спектр речи, фундаментальная частота, озвучивание и длительность моделируются одновременно по скрытой марковской модели. [11]
- Обучение моделей и генерация сигналов происходит по универсальному критерию максимального правдоподобия.

[12]

1.2.1. Преимущества генерации на основе CMM

- Статистическая параметрическая модель может быть эффективно обучена на речевых данных в цифровом формате с соответствующими транскрипциями.
- Статистическая модель на основе CMM имеет достаточно малый размер и эффективна в плане использования данных.

1.3. Генерация речи на основе глубинного обучения

Глубинное обучение оказало огромное влияние на исследования, продукты и сервисы в области автоматического распознавания речи. Перечислим причины популярности использования глубоких нейросетевых архитектур в данной области.

1. Встроенное извлечение признаков.

Трудно спросить, что автоматическое или хотя бы частично-автоматическое извлечение признаков просто мечта для исследователя. Особенно при работе с такими сложными распределениями как изображения и звук, когда внутреннее устройство признаков в значительной мере представляет из себя чёрный ящик.

- Возможность эффективно моделировать сильно коррелирующие признаки большой размерности.
- Слоёная архитектура с нелинейными операторами позволяет бесплатно интегрировать извлечение признаков в языковую модель.

2. Распределённое представление.

Здесь имеется ввиду, что нейросети зачастую способные единовременно захватывать большой участок пространства данных, когда, к примеру, деревья решений дробят пространство данных на области, что приводит к высокой фрагментации.

- Может быть экспоненциально более эффективно чем фрагментированное представление.
- Лучшие описательные качества с меньшим количеством параметров.

3. Слоёная иерархическая структура системы генерации речи.

Глубокие нейросети неявно расслаивают промежуточные представление, при движении вглубь по слоям.

`Звуковая волна` → `концептуальное представление` → `лингвистическое представление` → `представление с упором на описание методов произношения` → `звуковая волна`. [19].

Хоть целью работы и не является генерация речи напрямую, может возникнуть вопрос, почему была выбрана та ли иная подлежащая нейросетевая архитектура. Благо создатели нейросетевой архитектуры WaveNet позиционируют её как новый state of the art, поэтомоу с любыми вопросами относительно качества генерации мы можем отправить в оригинальную статью [13].

Однако, качество генерации было не единственным критерием выбора, поэтому мы опишем предыдущую "state of the art" систему и обоснуем свой выбор.

1.3.1. Глубинные архитектуры для генерации речи

- HMM-DBN (USTC/MSR [8], [9]).
- DBN (CUHK [7]).
- DNN (Google [18]).
- DNN-GP (IBM [5]).

Поскольку уместить обзор всех существующих решений в рамках данной работы не получится, мы сфокусируемся на предыдущем решении, позиционирующим себя как state of the art. Покажем преимущества нашего подхода и обоснуем свой выбор.

1.3.2. Deep Speech

- Использует рекуррентные нейронные сети для предсказания символьных последовательностей и потом применяет языковую модель.
- Использует 5 миллиардов связей для фразы `(utterance, x_t)` средней длины.
- Каждый GPU считывает порции фраз одинаковой или почти одинаковой длины.

- Для рекуррентных слоев один GPU идет слева-направо, другой справа-налево и посередине они обмениваются коэффициентами и меняются ролями.
- 5000 часов речи от 9600 людей перемешали с различным шумом и между собой и получили 100 000 часов данных для обучения.

[3]

1.4. Сравнение с существующим решением

Но даже Deep Speech не подходит для нашей задачи сразу по ряду причин:

1. Вычислительная сложность.

Как мы видим из описания Deep Speech рекуррентная сеть, главная особенность которой эффективное использование многих GPU, что требует огромных вычислительных ресурсов, которых у нас не было.

2. Естественность голоса.

Ключевой заявленной сильной стороной Wavenet является именно естественность генерируемого голоса. Авторы даже ввели специальную метрику, основанную на мнении независимых слушателей. Учитывая, что мы хотим генерировать особенные голоса, для нас такая черта подлежащей модели достаточно важна.

3. Нет инструментария для генерации с особенностями.

Ну и самое главное: неясно, как в такой архитектуре можно реализовать генерацию с особенностями.

2. WaveNet

WaveNet - это нейросетевая архитектура, вдохновлённая недавними достижениями в нейросетевых авторегрессивных генеративных моделей, описывающих сложные распределения такие как картинки [10] и текст [4]. WaveNet - это модель для генерации аудио, основанная на PixelCNN [2]. В статье аудио сигналы получаются с помощью генеративной модели, оперирующей напрямую с необработанным цифровым сигналом, описывающим звуковую волну.

Совместная вероятность волны $x = \{x_1, \ldots, x_T\}$ описывается как произведение условных вероятностей уравнением 1:

$$p(x) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}) \tag{1}$$

Таким образом, каждая точка аудио сигнала x_t зависит от всех предыдущих.

Определение 2.1 ***Стэком*** *слоёв в нейроести будем называть последовательный набор слоёв со следующим свойством: выход каждого слоя передаётся в качестве входа к следующему за ним. В контексте данной работы мы будем чаще всего говорить о стеках свёрточных слоёв.*

Определение 2.2 *В контексте свёрточных нейронных сетей* ***фильтром*** *называют набор обучаемых весов. Фильтр представляется в виде вектора, который участвует в свёртке с входными данными. Фильтры получили своё называние по аналогии с фильтрами, используемыми в цифровой обработке сигналов, и предоставляют меру сходства между входными данными и некоторым признаком. Признаки, которое помогают описать фильтры не проектируются явно, а извлекаются из данных как побочный результат алгоритма обучения.*

Определение 2.3 ***Рецептивное поле*** *в свёрточной сети это часть данных, которая "видна" фильтру в момент времени. Размер рецептивного поля растёт линейно относительно размера стэка обычный свёрток и экспоненциально для дырявых свёрток. Об этом подробнее в секции про свёртки. 2.1*

Так же как и в PixelCNN [2] распределение условной вероятности моделируется с помощью стэков свёрточныйх слоёв. В архитектуре нет слоёв пулинга и выход модели имеет такую же размерность что и вход. Модель выдаёт категориальное распределение над последующими значениями x_t с помощью softmax слоя. Модель оптимизирована выдавать логарифм функции правдоподобия данных с учётом параметров. Поскольку мы можем следить за логарифмом правдоподобия, мы также способны настраивать гиперпараметры на валидационном множестве, что позволит регулировать недообучение/переобучение.

2.1. Свёртки

Свёртки - главный ингредиент WaveNet. Используя свёртки, мы гарантируем, что модель не может нарушить первоначальный порядок данных: предсказание $p(x_{t+1} = x_1, \ldots, x_t)$, выдаваемое моделью в момент времени t не может зависеть ни от каких моментов в будущем $x_{t+1}, x_{t+2}, \ldots, x_T$. Вы это можете наблюдать на рисунке 2.

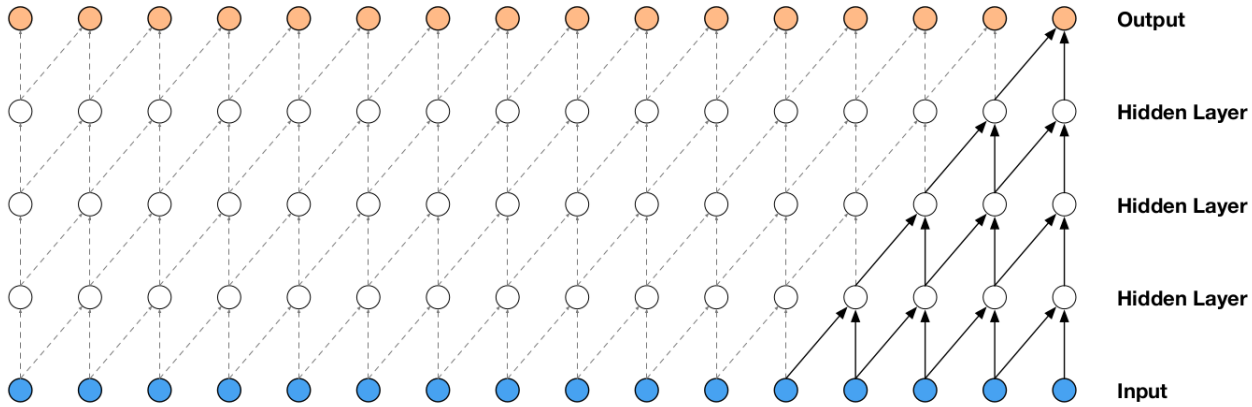

Рис. 2: Рецептивное поле для стэка свёрточных слоёв

При обуении условные вероятности во все моменты времени могут вычисляться параллельно, потому что все значения сигнала x заранее известны. Во время генерации предсказания последовательны: каждая предсказанная точка подаётся обратно на вход нейросети, чтобы предсказать следующую.

Поскольку модели со свёртками не имеют рекуррентных соединений, они обычно обучаются быстрее нежели рекуррентные архитектуры, особенно когда применяются к длинным последовательностям. Одна из проблем свёрток заключается в том, что они требуют большого количества слоёв либо большие размеры фильтров, чтобы увеличить ширину рецептивного поля. Например, на рисунке 2 ширина рецептивного поля равна 5 (=#слоёв + длина фильтра - 1). Для решения этой проблемы и появились свёртки с дырками, позволяющие на порядки увеличить рецептивное поле, не накладывая больших вычислительных затрат.

2.2. Дырявые свёртки

Дырявые свёртки это свёртки, в которых фильтр применяется по диапазону больше своей длины, пропуская входные значения с некоторым шагом. Это эквивалентно свёртке с большим фильтром, "продырявленным" нулями, но значительно эффективнее вычислительно. Такая эффективность дырявых свёрток позоволяет нейросети оперировать с более крупным данными, нежели позволили бы обычные свёртки. В частном случае дырявые свёртки с пропуском 1 эквиваленты обычным свёрткам. На рисунке 3 изображены дырявые свёртки с промежутками 1, 2, 4 и 8.

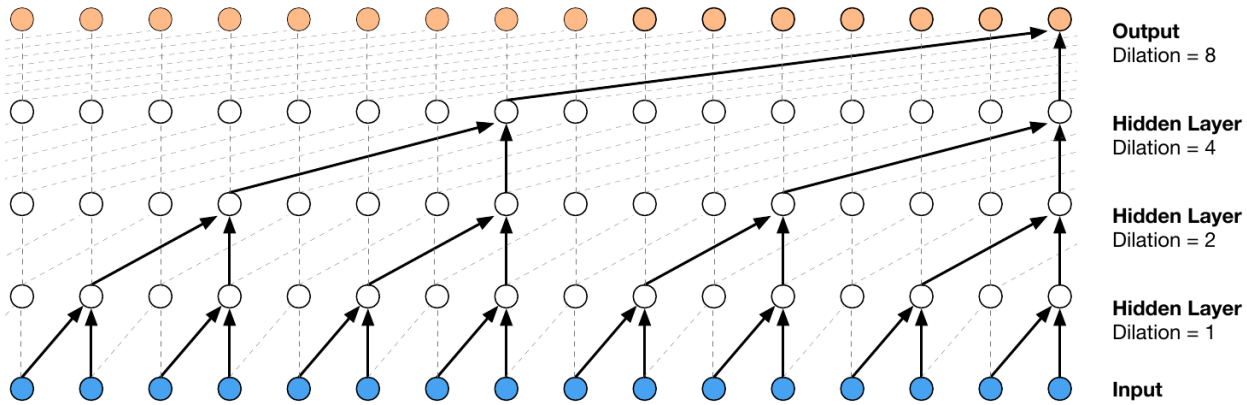

Рис. 3: Рецептивное поле для стэка дырявых свёрточных слоёв

Стэки дырявых свёрток позоволяют достичь очень широкого рецептивного окна с помощью небольшого количества слоёв, при этом сохраняя качество входных данных на протяжении всей сети. В оригинальном WaveNet промежутки увеличиваются в два раза для каждого слоя пока не достигнут какого-то фиксированного значения, и затем всё повторяется. Например:

$$1, 2, 4, \ldots, 512, 1, 2, 4, \ldots, 512, 1, 2, 4, \ldots, 512.$$

Попробуем объяснить интуицию за такой конфигурацией. Во-первых, экспоненциальное увеличение промежутков приводит к экспоненциальному по глубине росту окна. [17] Например, каждый $1, 2, 4, \ldots 512$ блок имеет окно размера 1024 его можно воспринимать в качестве более эффективной и имеющей большую описательную силу альтернативы свёртки 1×1024. Во-вторых, дальнейшее объединение таких слоёв в стэки увеличивает объём модели и ширину окна.

2.3. Gated activation unit

Определение 2.4 ***Gate*** *в контексте нейросетей это термин, мигрировавший из электрических сетей. Поскольку любые операции могут быть описаны в качестве графовых примитивов, gate в самом широком смысле можно назвать операцию любой арности. Одним из простейших примеров gate является умножение.*

Ключевой gate в архитектуре называемый update gate находится на выходе из каждого пачки свёрточных слоёв.

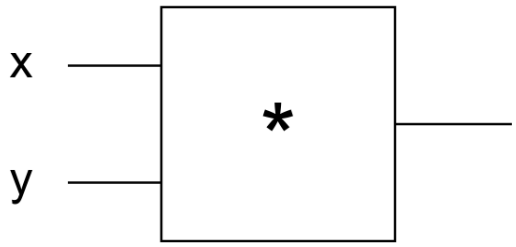

Рис. 4: Простейший gate

Gated activation unit выглядит так:

$$z = \tanh(W_{f,k} * x \odot \sigma(W_{g,k} * x)) \tag{2}$$

где $*$ обозначает операцию свёртки, $\odot$ обозначает поэлементное умножение, $\sigma(\cdot)$ сигмоида, k номер слоя, f и g обозначают фильтр и gate соответственно, и W обучаемый свёрточный filter. Авторы WaveNet утверждают, что такая функция активации работает значительно лучше для генерации аудиосигналов чем стандартная rectified linear activation function.

2.4. Residual и skip соединения

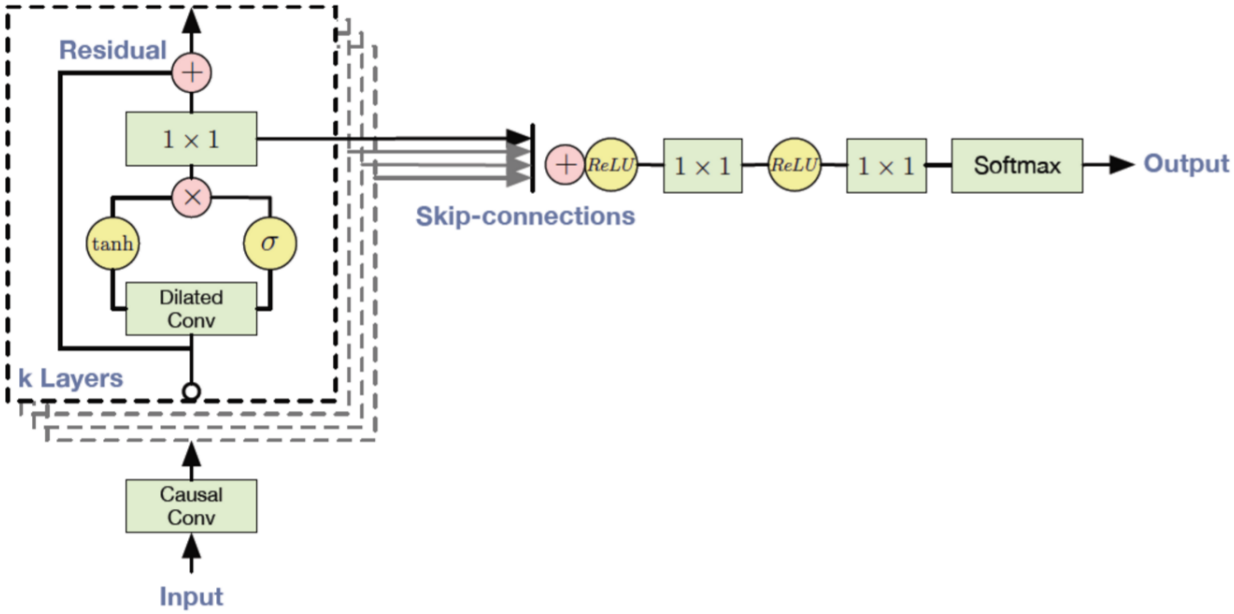

Рис. 5: Архитектура WaveNet

Определение 2.5 ***Skip*** *соединение в нейронной сети это соединение, которое пропускает слой и соединяется к следующему доступному слою. В общем случае может быть пропущен более чем один слой.*

Определение 2.6 ***Residual*** *соединение присоединяется к предыдущему слою.*

В архитектуре используются residual и параметрические skip соединения, чтобы ускорить сходимость и позволить обучение более глубоких моделей. На рисунке виден residual блок модели, который повторяется много раз в архитектуре.

2.5. WaveNet с условием

Если дан дополнительный вход h как условие, WaveNet способен моделировать условное распределение $p(x|h)$ аудио по этому входу. Уравнение 1 теперь принимает вид 3:

$$p(x|h) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, h) \tag{3}$$

Задавая такого рода условия, мы сподвигаем WaveNet к генерации аудио с необходимыми характеристиками. Например, если наши исходные данные для обучения содержат большое количество ораторов, мы можем передавать идентификатор говорящего в качестве глобального условия. Тогда впоследствии при генерации мы сможем выбирать, чью речь мы хотим сгенерировать.

2.5.1. Локальные и глобальные условия

Архитектура WaveNet предоставляет два способа передачи условий: глобальное условие и локальное условие. Глобальное условие характеризуется единственным скрытым представлением $\mathbf{h}$, которое оказывает влияние на финальное распределение вдоль всего временного промежутка. Функция активации из формулы 2 теперь принимает вид 4:

$$z = \tanh(W_{f,k} * x + V_{f,k}^T h \odot \sigma(W_{g,k} * x + V_{g,k}^T h)). \tag{4}$$

где $V_{*,k}$ это свёртка 1×1. В качестве альтернативы такой миниатюрной свёрточной сети можно использовать $V_{f,k} * h$ и продублировать это значения по времени. Однако, авторы утверждают, что такой подход работает гораздо хуже на практике.

3. Основные цели

Сформулируем цели, которые мы ставим в этой работе:

1. Реализовать WaveNet максимально придерживаясь описания из статьи.

    - Реализовать генерацию голоса без условия.
    - Реализовать генерацию голоса по тексту.
    - Реализовать генерацию голоса по тексту с условием.

Важно отметить, что ключевым аспектом этой подзадачи является педантичная точность в реализации архитектуры сети. В оригинальной статье архитектура описана довольно высокоуровнево и в общих словах, что подразумевает от читателя уверенных знаний в области и навыков проектирования сетей.

Нам также требуется реализовать дополнительные модификация для локальных и глобальных условий, поскольку они необходимы для генерации с особенностями.

2. Разработать признаки для генерации голоса

Требуется спроектировать, реализовать и провести апробацию набора признаков, передаваемых в сеть по каналам локальных и глобальных условий. Авторы WaveNet бегло упоминают используемые признаками, оставляя эту задачу пользователям.

3. Получить результаты генерации

Наша постановка целей сформулирована так, что мы не можем описать качество работы модели в виде численных измерений. Поэтому постараемся добиться высоких качеств натуральности голоса в результате экспериментов и получить примеры сгенерированной речи.

4. Методы и реализация

4.1. Описание данных

В качестве данных для обучения в данной работе использовался корпус CSTR VCTK [16], размещённый в открытом доступе. Корпус CSTR VCTK включает в себя речь, произнесённую 109 нативными носителями английского языка с разнообразными акцентами. Каждый оратор зачитывает около 400 предложений, выбранных из газет, также специальные отрывки, подобранные чтобы подчеркнуть акцент говорящего.

К каждой голосовой записи прилагается текст, не выровненный по аудио. То есть каждый пример из обучающегося множества представлен в виде пары (аудио, текстовый файл). Аудио прдесавленно в формате .wav с частотой сэмплирования 48000 Гц и одним каналом, то есть моно. Текст - обычный .txt файл без выравнивания по времени, в противном случае это были бы субтитры.

Все отрывки речи были записаны с использованием одинакового оборудования, предоставленного университетом Эдинбурга. Также использовалось специальное помещение, поэтому качество записи достаточно высокое и практически не требует фильтрации шумов.

Продолжительность аудио варьируется от 1 до 11 секунд со стандартным отклонением около секунды, что более чем достаточно для нашей задачи, так как генерация даже небольших звуковых файлов занимает достаточно много машинного времени в силу скромности наших вычислительных ресурсов.

4.2. Реализация модели

Мы старались реализовать практическую часть так, чтобы она принесла максимальный вклад в сообщество и не затерялась в качестве приложения к исследовательской работе. Среди открытых реализаций не нашлось ни одной, котрая в полной мере реализовывала бы все заявленные архитектурные нюансы WaveNet. Поэтому было решено доработать архитектуру в виде ответвления от самого известного в сообществе решения.

Таким образом, реализацию можно разделить на две большие части: модификацию архитектуры сети и извлечение построение каскада признаков признаков. Также стоит удостоить внимания постановку экспериментов, но мы уделим им отдельный раздел 5.

Срез архитектуры сети до модификаций имеет структуру, изображенную на рисунке 6.

Реализация WaveNet, описываемая в этом разделе, реализована на языке Python на фреймворка для численных графовых вычислений с открытым исходным кодом

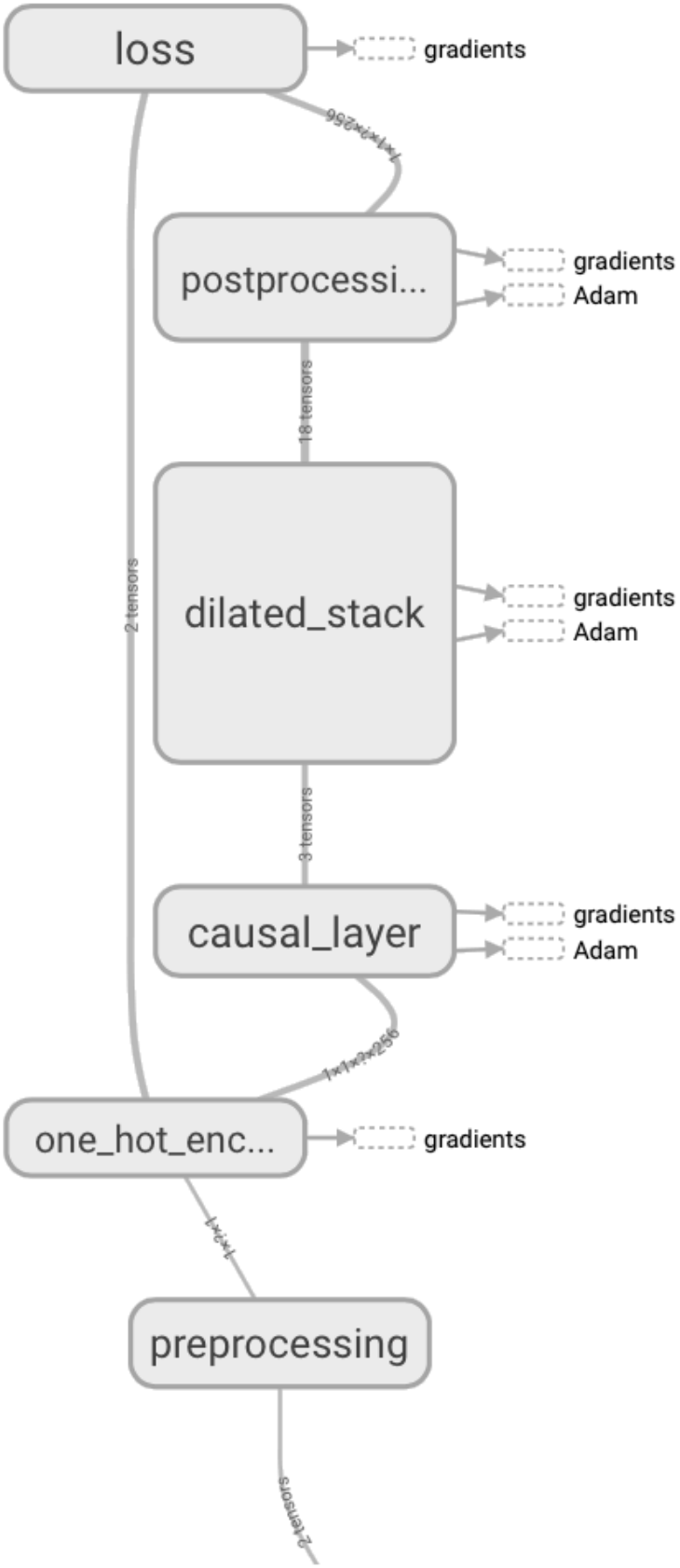

Рис. 6: Высокоуровневое описание архитектуры сети

Tensorflow. На момент написания статьи Tensorflow удерживает позицию одного из самого популярного фреймворков для проектирования и промышленной разработки нейронных архитектур. По ходу раздела мы будем переключатся между абстрактным описанием WaveNet и обозначениями в вычислительном графе конкретной реализации, однако я постараюсь поддерживать единообразные наименования, чтобы не вызывать у читателя лишнего дискомфорта.

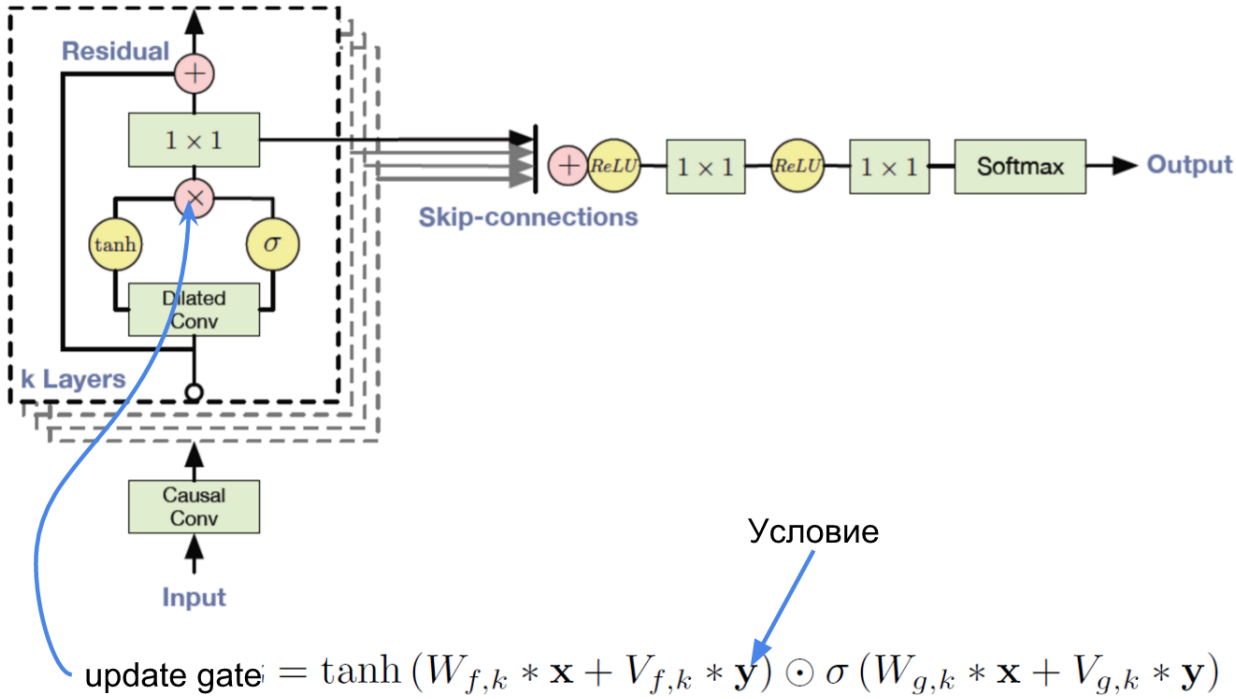

Рис. 7: Gate, отвечающий за условия

На рисунке 7 заметно, что в архитектуре дублируются k раз одинковые конструкции которые мы отныне будем называть **dilated stack**. Из каждого dilated stack выходит skip соединение на активационный gate и residual соединение в самого себя. Основной частью dilated stack является стэк свёрточных слоёв единообразной конфигурации, задаваемой в гиперпараметрах сети.

Как мы уже знаем, дополнительные условия передаваемы WaveNet должны быть переданы в update gate 7. А так как такой gate есть в каждом dilated stack, мы на самом деле должны продублировать эти условие тоже k раз.

В архитектуре это выглядит почти симметрично для локальных и глобальных условий с точностью до размерности тензоров. Мы должны добавить пару (`gc_filter`, `gc_gate`) и (`lc_filter`, `lc_gate`) для глобальных и локальных условий соответственно, чтобы потом передать результат свёртки с ними на update gate. Для глобальных условий на фиксированном слое это изображено на рисунке 8.

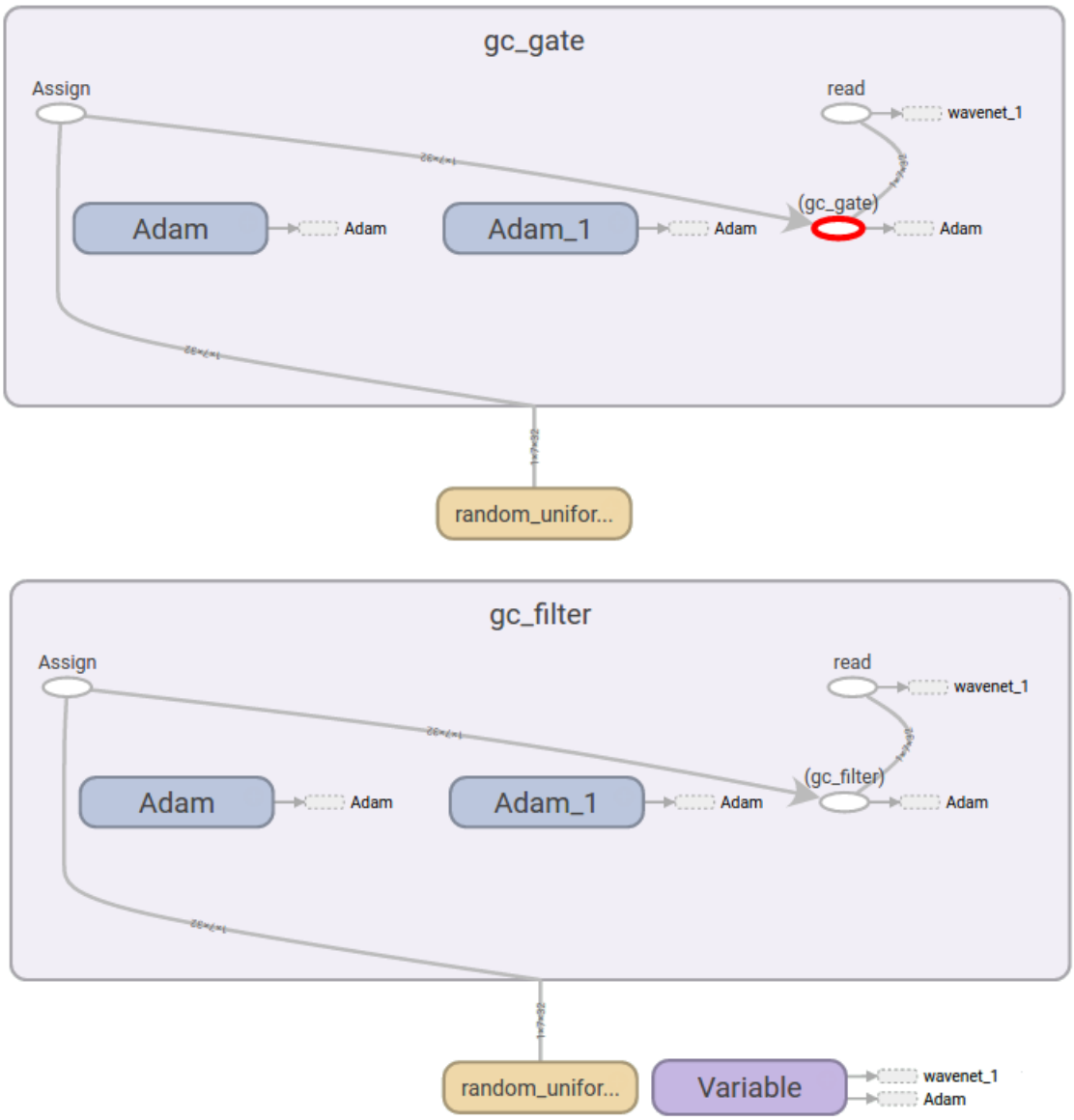

Рис. 8: Глобальные условия в архитектуре

4.2.1. Обработка условий

Давайте ещё раз вспомним, по каким каналам данные поступают в WaveNet:

- Сырые данные.
  - Временной ряд, цифровое представление голоса по времени.
- Локальное условие .
  - Временной ряд, той же длины что и данные. Качество, изменяющееся по времени.
- Глобальное условие.
  - Качество говорящего, не зависящее от времени. Не меняет своего значения в процессе обучения/генерации.

Прежде чем быть переданными на вход нейросети данные должны имть чётко структурированное векторизованное представление. Однако недостаточно просто закодировать данные. На самом деле мы провели преобразование данных в векторный вид ещё не стадии извлечения признаков, что порождает вопрос, почему бы и не передавать это представление в качестве условий. На самом деле это было бы невозможно даже с точки зрения архитектуры. Ещё во вводной части мы упоминали, что условия должны быть выровнены вдоль данных, то есть иметь ту же длину. Если говорить о звуке как о временном ряде, то временной ряд условий должен иметь ту же длину, что и голос, однако на ширину архитектура ограничений не налагает.

4.2.2. Глобальные условия

В первую очередь опишем проблему построения глобальных условий, поскольку она решается проще. С точки зрения временного ряда глобальное условие это значение, константное во все моменты времени. Поэтому для глобальных условий достаточно продублировать значение признака необходимое количество раз.

4.2.3. Локальные условия

С локальными условиями дело обстоит сложнее, поскольку разные признаки могут иметь разную длину. Каждый признак требует индивидуального ad hoc решения. Это связано с тем, что локальное условие должно не только иметь требуемую архитектурой длину, но и быть актуальным в момент аудио, которому сопоставлено.

Разберём на примере одного признака. Наиболее сложный случай возникает когда признаки извлекаются не из звукового отрезка, а из соответствующего ему текстового

файла. Мало того что у нас текст не выровнен по звуку, так ещё и из этого текста мы извлекаем признаки, получая двойную погрешность в выравнивании.

В качестве примера опишем достаточно интуитивный пример локального условия, основывающийся на тексте. На рисунке 9 изображено, какой путь проходит текст, прежде чем попасть в `dilated_stack`.

$$\texttt{one\_hot\_1} \to \texttt{ExpandDimention} \to \texttt{Resize} \to \texttt{Squeeze}.$$

В качестве кодирования будем хэшировать каждый символ, после чего растягивать полученный ряд вдоль звука. Интуитивный эквивалент такого преобразования будет выглядеть так:

$$\texttt{'The car'} \to \texttt{'TTTTTTThhhhhheeeeee ccccaaaarrrr'}.$$

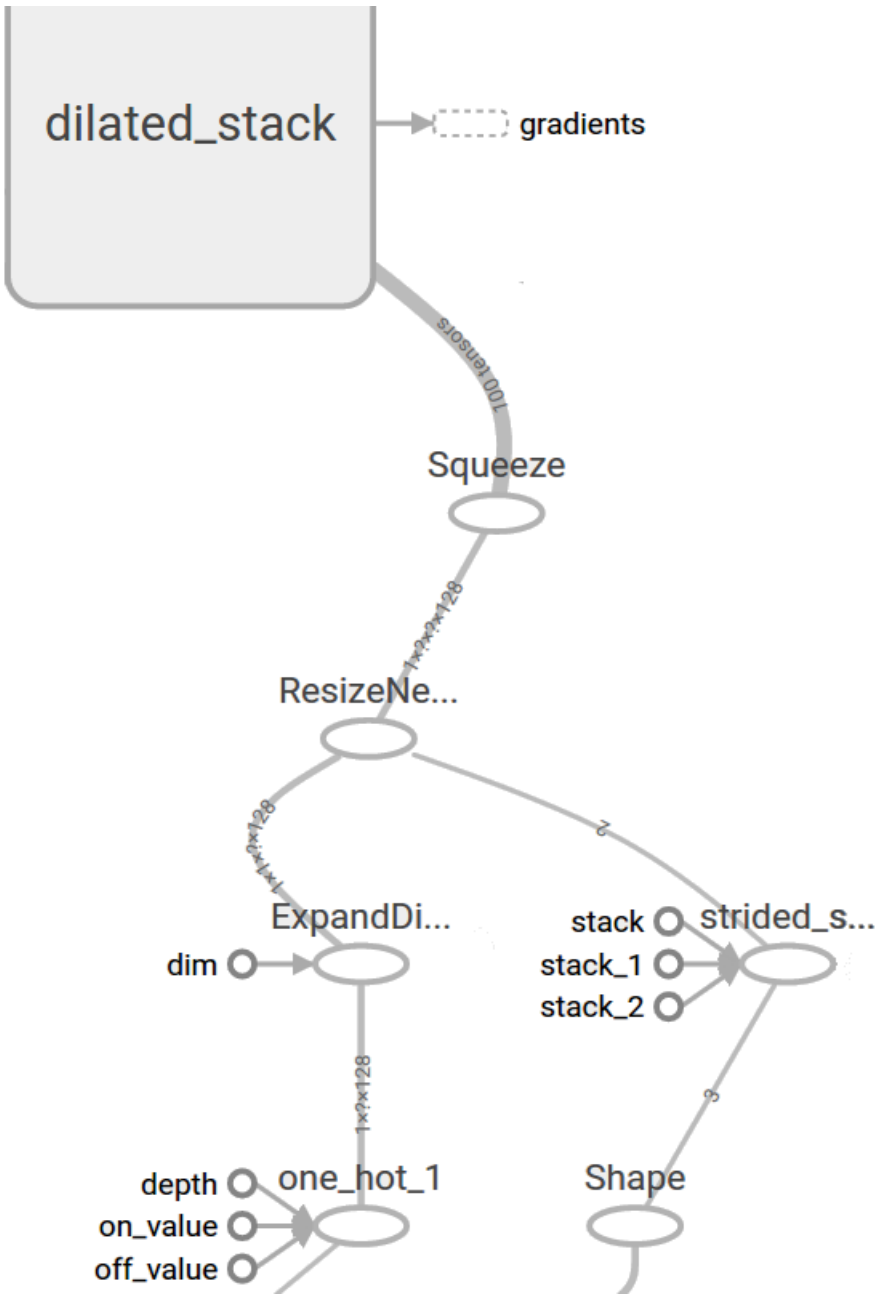

Рис. 9: Обработка локальных текстовых условий

4.3. Извлечение признаков

В предыдущем подразделе мы уже описали пример признака, извлекаемого в качестве локального условия. Стоит упомянуть о том, что проектирование и подбор оптимального набора признаков это своего компромисс. Дело в том, что время обучения WaveNet растёт линейно от ширины слоёв, которые в свою очередь линейно зависят от количества признаков. Ввиду наших достаточно скромных для такой задачи как генерация голоса вычислительных ресурсов, мы лишены роскоши использования настолько большого набора признаков, насколько мы способны реализовать. Несмотря на это, в этом разделе мы опишем признаки, которые мы посчитали полезными

для повышения качества генерации, однако тем, которые использовались в качестве локальных условий, уделим отдельное внимание.

4.3.1. Признаки для представления данных

Опишем признаки, которые извлекаются нашим скриптом из звукового файла таблицей 1.

Таблица 1: Голосовые признаки

Id	Признак	Описание
1	Zero Crossing Rate	Количество смен знака сигнала в рамках временного фрейма
2	Energy	Сумма квадратов значений сигнала, нормализованная по длине фрейма
3	Entropy of Energy	Энтропия нормализованной энергии подфреймов
4	Spectral Centroid	Центр тяжести спектра
5	Spectral Spread	Второй центральный момент спектра
6	Spectral Entropy	Энтропия нормализованных энергий спектра для набора подфреймов
7	Spectral Flux	Квадратичное отклонение между нормализованными аплитудами спектров двух соседних фреймов
8	Spectral Rolloff	Частота ниже которой сконцетрированны 90% значений распределения
9-21	MFCCs	MEL-кепстральное коэффициенты представления
22-33	Chroma Vector	12-элементное представление спектральной энергии, элементы которого представляют 12 равноценных классов тангажа музыки западного типа
34	Chroma Deviation	Стандартное отклонение предыдущих 12 коэффициентов

Каждый признак извлекается фреймами по 10 миллисекунд без пересечений, но также дублируется на фреймах по 100 миллисекунд с шагом 10 миллисекунд, то есть с пересечениями, причём с заглядыванием как вперёд, так и назад.

Однако такой набор признаков весьма сильно нагружает канал локального условия, заставляя сеть сходиться очень долго. Так как эксперименты показали, что сеть может генерировать приличный голос даже без них, решено было в финальном эксперименте от дополнительных признаков для представления данных отказаться.

4.3.2. Признаки для глобального условия

Признаки для глобального условия должны описывать некоторую характеристику, которая будет задавать особенность генерируемой моделью речи. В финальных экспериментах использовались два основных признака: пол и идентификационный номер говорящего. Для того чтобы использовать идентификационный номер на этапе

генерации, нужно задать ещё на стадии обучения, сколько всего в обучающей выборке разных голосов, чтобы в реализации построить `one-hot encdoing` для номеров говорящих.

4.3.3. Признаки для локального условия

Признаки для для локального условия должны в основном опираться на текст, поскольку мы стараемся добиться, чтобы генерируемая речь была более осмыслена и в перспективе приближала модель к решению задачи трансляции текста в голос. В таком случае стандартный word2vec автоэнкодер нам совсем не подходит, потому что последний улавливает лишь семантическую близость между словами, а мы хотели бы иметь высокоуровневое представление, содержащее больше информации о том, **как** произносить слова.

Первым приходящим на ум представлением, объединяющим в себе морфологические свойства речи были бы морфемы. Однако вспомним, что в силу нюансов архитектуры всплывёт двойная проблема с выравниваем: выравнивание морфем вдоль текста, а потом выравнивание результатов морфологического анализа вдоль звука. В такой постановке извлечение признака звучит очень рискованно и трудоёмко, в итоге мы рискуем получить сильную погрешность за счёт промежуточных шагов, потратив много усилий на нетривиальную реализацию.

Альтернативным решением стало использовать в качестве признаков речь, сгенерированную открытым языковым веб-интерфейсом Yandex Speech Kit [20]. Для этих был выбран yandex-speech-kit по двум основным причинам

1. Speech-kit имеет бесплатный суточный лимит для исследовательних целей
2. Yandex любезно согласился расширить для нас этот лимит, поскольку у нас очень много данных

Для генерации мы использовали голос по умолчанию Jane. Поскольку в корпусе VCTK очень часто используются одни и те же фразы, мы завели глобальную хеш-таблицу всех фраз, чтобы запоминать, для каких фраз мы уже сгенерировали речь и переиспользовать.

Такое представление достаточно высокоуровневое и компактное, что позволит нам долждаться результатов обучения.

4.3.4. Выравнивание аудио

Теперь стоит рассказать, как мы избавились от проблемы с выравниванием. В общем случае выравнять два временных ряда на одинаковое количество фреймов это достаточно ad hoc задача, сильно зависящая от семантики извлекаемого признака. Основная проблема заключается в том, что в результаты выравнивания мы теряем

информацию для более длинного ряда либо эмулируем недостающую для более короткого. В нашем случае, нашёлся удобный способ обойти эту проблему, варьируя частоту сэмплирования при чтении аудио.

Обозначим длины оригинального и сгенерированного аудио как LEN_{orig} и LEN_{gen} соответственно при частоте сэмплирования = 16000 Гц. Тогда чтобы получить временные ряды одинаковой длины считаем ещё раз сгенерированное аудио отнормировав частоту на величину $\frac{LEN_{orig}}{LEN_{gen}}$. Если у нас где-то возникнет погрешность на пару фреймов из-за деления, исправим их механически, просто выкинув лишние из конца.

5. Обсуждение результатов

Рис. 10: Сравнение сгенерированного сигнала с сигналом из обучающей выборки

Мы провели апробацию нашей модели в разных конфигурациях, с выбором признаков и подмножеств данных. Результаты можно увидеть в таблице 2

Результаты генерации сложно описать в терминах численных характеристик, так как основным критерием качества мы считает естественность сгенерированного голоса. В этом мы солидарны с авторами WaveNet, которые оценивали качество своих результатов с помощью некоторого Mean Optiaml Score, который по сути основывался на оценках слушателей по пятибальной шкале (1: Плохо, 2: Слабо, 3: Удовлетворительно, 4: Хорошо, 5: Отлично). К сожалению, мы не можем совсем честно поставить эксперимент и получить оценку независимых слушателей, поэтому придётся поверить оценке автора диссертации.

5.1. Сравнение качества результатов

Для оценки качества независимым испытуемым проигрывались пары двухсекундных сгенерированных промежутков в двух разных конфигурациях и слушатель должен был сделать выбор, какая запись кажется ему более натуральной либо же завить, что варианты для него индифферентны. Для сравнения мы использовали конфигу-

Таблица 2: Результаты экспериментов

Конфигурация WaveNet	**Данные**	**Результат**
Без модификаций	весть корпус	речеподобный звук
Без модификаций	одна фраза, много голосов	высокий гул
Без модификаций	один голос, много фраз	шум
Глобальное условие: ID говорящего	весь корпус	речеподобный звук
Глобальное условие: ID говорящего	одна фраза	шум
Глобальное условие: пол говорящего	два голоса, много фраз	шум
Глобальное условие: пол говорящего	два голоса, одна фраза	шум
Локальное условие: текст	весь корпус	шум
Локальное условие:текст	одна фраза	шум
Локальное условие: yandex-speech	весь корпус	речеподобный звук лучшего качества
Локальное условие: yandex-speech	одна фраза	высокий гул
Глобальное условие ID говорящего + локальное условие yandex-speech	весь корпус	шум
Глобальное условие ID говорящего + локальное условие yandex-speech	одна фраза	шум

Таблица 3: Сравнение натуральности генерируемых голосов

Субъективное предпочтение(%) в натуральности голоса			
Без модификаций на всём корпусе	С глобальным условием ID говорящего на всём корпусе	С локальным условием yandex-speech на всём корпусе	Нет предпочтения
27		**61**	11
34	**34**		31
	29	**56**	15

рации, показавшие лучшее качество генерируемого звука. Результаты представлены в таблице 3.

5.2. Производительность

Мы считали на GPU Tesla K80 с 11 гигабайтами видеопамяти с поправками на виртуализацию.

Чтобы добиться значений хотя функции потерь хотя бы как на рисунке нам требуется около 4 суток. Генерация пяти секунд аудио занимает около часа.

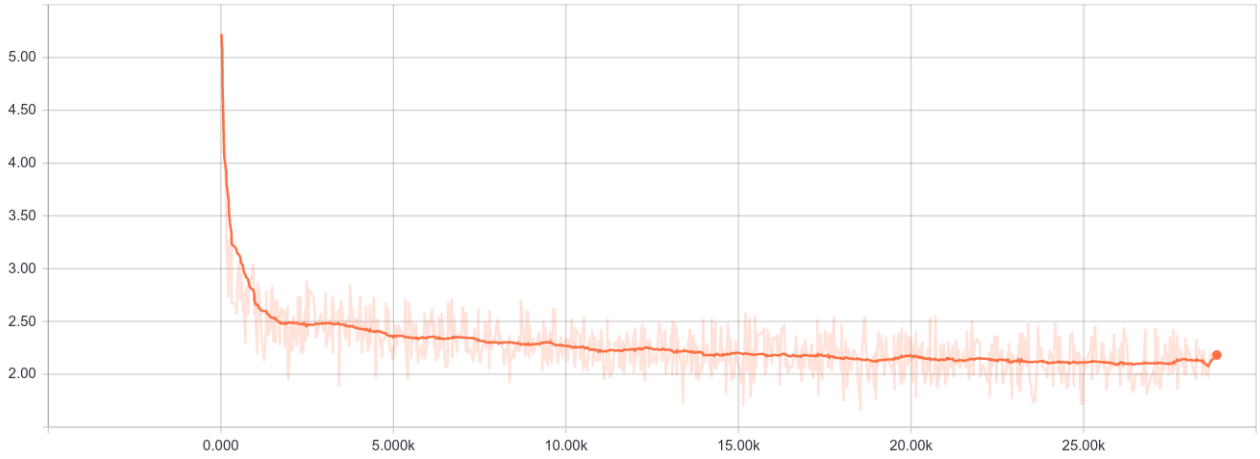

Рис. 11: Изменение функции потерь в процессе обучения

5.3. Анализ экспериментов

Исходы экспериментов можно просуммировать следующим образом:

1. WaveNet реализован корректно.

Это видно из экспериментов на модели без модификаций. Также показываем, что описательной силы архитектуры на сырых данных хватает, чтобы описать распределение порождаемое звуковым сигналом.

2. Глобальные условия снижают качество.

Почти во всех экспериментах с глобальными условиями не получилось хороших результатов.

3. Использование результатов генерации независимой системы оправдывает себя в качестве локального условия.

Модель с этими настройками даёт лучшие показатели натуральности голоса, из тех, что нам удалось добиться. Однако хотелось бы добавить больше признаков, поскольку генерируемой речи не хватает структуры.

4. Недостаточно качественных текстовых признаков.

Тесты показали, что используемые нами текстовые признаки сказываются на качестве генерации отрицательно. Нам кажется, что основная проблема заключается в неточностях выравнивания и такие данные только запутывают модель. Возникла гипотеза, что с этой проблемой может помочь справиться реализация механизма внимания.

5. Модель требует больше времени чтобы сойтись.

Возможно, требуется больше времени для обучения сети, потому что как мы видим на рисунке функция потерь не перестаёт падать, просто мы не можем себе позволить ждать так долго. Однако такая гипотеза вызывает сомнения, потому что переобученные модели давали на выходе к очень высокий неестественный звук.

6. Заключение

Каких результатов мы добились:

1. Реализован WaveNet максимально придерживаясь описания из статьи.

   - Доработана существующая генерация голоса без условия.
   - Реализована генерация голоса по тексту.
   - Реализована генерация голоса по тексту с условием.

Мы дополнили публичную реализацию WaveNet с максимальной точностью повторяя описание из статьи, а также реализовали нюансы, опущенные в оригинальной статье, но важные для удобства использование модели. Одним из таких нюансов стало выравнивание длин локальных условий.

2. Разработаны признаки для генерации голоса.

Основным достижением в этом пункте мы считаем реализацию и апробацию набора признаков, основывающихся на сгенерированной речи худшего качества сгенерированной другим инструментом. Таким образом мы не только нашли легковесное высокоуровневое представление для локальных условий, но и неявно предоставили механизм для комбинации WaveNet с другими системами для генерации речи.

3. Получены результаты генерации.

Мы получили порядка двадцати примеров сгенерированной речи на базе нашей модели с разными конфигурациями и сделали выводы о применимости рассмотренных подходов.

6.1. Направления развития

1. Приспособить модель для решения задачи text2speech.

Теперь, когда у нас есть полностью модифицированный WaveNet с некоторым каскадом дополнительных признаков и локальных условий, мы можем нацелиться на задачу генерации речи по тексту. В таком контексте стоит обратить больше внимания текстовым признакам и сместить центр внимания от натуральности генерируемой к способности модели произносить тексты.

2. Реализовать более ярко выраженную условную генерацию.

В рамках экспериментов не удалось бесплатно достичь значимой разницы в речи при условной генерации за счёт включения простых глобальных условий. Основной причиной тому послужило падение качества речи при включении глобальных условий. Исходя из этого, у работы есть ещё и такой вектор развития в сторону более яркого контраста на базе глобальных условий.

Список литературы

[1] Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. / Jeremy Elson, John R Douceur, Jon Howell, Jared Saul // ACM Conference on Computer and Communications Security / Citeseer. — Vol. 7. — 2007. — P. 366–374.

[2] Conditional Image Generation with PixelCNN Decoders / Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals et al. // arXiv preprint arXiv:1606.05328. — 2016.

[3] Deep speech: Scaling up end-to-end speech recognition / Awni Hannun, Carl Case, Jared Casper et al. // arXiv preprint arXiv:1412.5567. — 2014.

[4] Exploring the limits of language modeling / Rafal Jozefowicz, Oriol Vinyals, Mike Schuster et al. // arXiv preprint arXiv:1602.02410. — 2016.

[5] F0 contour prediction with a deep belief network-Gaussian process hybrid model / Raul Fernandez, Asaf Rendel, Bhuvana Ramabhadran, Ron Hoory // Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on / IEEE. — 2013. — P. 6885–6889.

[6] Kaggle. Dogs vs. Cats. — 2017. — URL: `https://www.kaggle.com/c/dogs-vs-cats` (online; accessed: 01.08.2017).

[7] Kang Shiyin, Qian Xiaojun, Meng Helen. Multi-distribution deep belief network for speech synthesis // Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on / IEEE. — 2013. — P. 8012–8016.

[8] Ling Zhen-Hua, Deng Li, Yu Dong. Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis // IEEE Transactions on Audio, Speech, and Language Processing. — 2013. — Vol. 21, no. 10. — P. 2129–2139.

[9] Ling Zhen-Hua, Deng Li, Yu Dong. Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis // Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on / IEEE. — 2013. — P. 7825–7829.

[10] Oord Aaron van den, Kalchbrenner Nal, Kavukcuoglu Koray. Pixel recurrent neural networks // arXiv preprint arXiv:1601.06759. — 2016.

[11] Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis / Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko et al. // Sixth European Conference on Speech Communication and Technology. — 1999.

[12] Tokuda Keiichi, Zen Heiga, Black Alan W. An HMM-based speech synthesis system applied to English // IEEE Speech Synthesis Workshop. — 2002. — P. 227–230.

[13] Wavenet: A generative model for raw audio / Aäron van den Oord, Sander Dieleman, Heiga Zen et al. // CoRR abs/1609.03499. — 2016.

[14] Wikipedia. Generative model // Wikipedia, the free encyclopedia. — 2017. — URL: https://en.wikipedia.org/wiki/Generative_model (online; accessed: 11.05.2017).

[15] Wikipedia. Hidden Markov model // Wikipedia, the free encyclopedia. — 2017. — URL: https://en.wikipedia.org/wiki/Generative_model (online; accessed: 11.05.2017).

[16] Yamagishi Junichi. CSTR VCTK Corpus // The Centre for Speech Technology Research. — 2016. — URL: http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html,urldate="01.08.2017",language="english".

[17] Yu Fisher, Koltun Vladlen. Multi-scale context aggregation by dilated convolutions // ICLR,.

[18] Ze Heiga, Senior Andrew, Schuster Mike. Statistical parametric speech synthesis using deep neural networks // Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on / IEEE. — 2013. — P. 7962–7966.

[19] Zen Heiga. Deep Learning in Speech Synthesis // Google. — 2013. — URL: https://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/41539.pdf,urldate="01.08.2017",language="english".

[20] Речевые технологии Яндекс // Яндекс. — 2013. — URL: https://tech.yandex.ru/speechkit/?ncrnd=1644,urldate="01.08.2017",language="russian".