УДК 004.89 **Информационные технологии**

Белоножко Павел Евгеньевич, *студент-магистр, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Россия, г. Калуга*

Белов Юрий Сергеевич, *к.ф. -м.н., доцент, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Россия, г. Калуга*

РЕАЛИЗАЦИЯ ВОКОДЕРА ДЛЯ СИСТЕМЫ ПРЕОБРАЗОВАНИЯ ТЕКСТА В РЕЧЬ НА ОСНОВЕ RNN МОДИФИКАЦИИ МОДЕЛИ WAVENET

Аннотация: Любая современная генеративная модель синтеза клонированной речи состоит из двух обязательных элементов: генератора текста в речь, с помощью которого происходит генерация звукового содержимого речи с последующим составление мел-спектрограммы, и вокодера, который получает на вход данные звукового содержимого речи в виде мел-спектрограммы и данные голоса говорящего с последующим синтезом речи этим голосом. Для первого элемента самой эффективной реализацией является модель, основанная на Tacotron-2. Основной проблемой в построении архитектуры любой TTS-системы является выбор вокодера для синтеза речи. Выбор оптимального варианта вокодера может зависеть от требований конкретной задачи, ресурсов, доступных для обучения и использования модели, а также от возможностей оптимизации производительности.

Ключевые слова: вокодер, преобразование текста в речь, TTS, WaveNet, WaveRNN, рекуррентная нейронная сеть, акустические особенности.

Abstract: Any modern generative model for synthesizing cloned speech consists of two mandatory elements: a text-to-speech generator, which generates the

sound content of speech with subsequent compilation of a chalk spectrogram, and a vocoder that receives data of the sound content of speech in the form of a chalk spectrogram as input and data of the speaker's voice, followed by speech synthesis by this voice. For the first element, the most efficient implementation is the model based on Tacotron-2. The main problem in building the architecture of any TTS system is the choice of a vocoder for speech synthesis. The choice of the optimal vocoder option may depend on the requirements of a particular task, the resources available for training and using the model, as well as performance optimization opportunities.

Keywords: vocoder, text-to-speech, TTS, WaveNet, WaveRNN, recurrent neural network, acoustic features.

Введение. На сегодняшний день наиболее популярной реализацией вокодера является модель WaveNet. WaveNet является глубокой генеративной моделью, которая использует сверточные нейронные сети для прямого синтеза речи из текста. Однако, WaveNet имеет довольно высокую стоимость генерации, так как требует большого количества вычислительных ресурсов и времени для генерации даже небольших аудио-сигналов. Поэтому для ускорения процесса генерации аудио сигналов была предложена модификация оригинальной модели WaveNet, которая имеет название WaveRNN. Она использует рекуррентные нейронные сети (RNN) вместо сверточных для генерации аудио-сигнала. Это позволяет более эффективно учитывать зависимости между соседними отсчетами звука, что позволяет достичь более высокой скорости генерации. Кроме того, WaveRNN также использует механизмы выборки (sampling) и квантования (quantization) для уменьшения числа параметров и ускорения процесса генерации. WaveRNN является более эффективной альтернативой WaveNet и может быть использован для быстрого и качественного синтеза речи [1].

Архитектура оригинальной модели WaveNet. Предполагается, что с есть входной сигнал x_t, который представляет собой аудио-сигнал с дискретизацией 16 кГц. Входной сигнал обрабатывается с помощью серии сверточных слоев,

каждый из которых имеет ядро свертки размера 2, 3 или 4. В каждом сверточном слое используется ReLU-активация для усиления нелинейности модели.

Пусть y_l обозначает выходные данные l сверточного слоя. Тогда y_l можно рассчитать следующим образом:

$$y_l = \sigma(W_l * h_{l-1} + b_l)$$

Здесь W_l - это матрица весов для l слоя свертки, символ умножения обозначает операцию свертки, h_{l-1} - это входные данные для l слоя, b_l- это смещение, а σ - это нелинейная функция активации (например, ReLU).

Однако, в отличие от обычных сверточных сетей, в WaveNet используется свертка с дилатацией, что означает, что в каждом слое используются ядра свертки, которые «разрежены» на определенном шаге. Предположим, что дилатация для l сверточного слоя равна 2^{l-1}.

Тогда y_l можно выразить следующим образом:

$$y_l = \sigma\left(W_l * h_{l-1}^{(l)} + b_l\right)$$

где $h_{l-1}^{(l)}$ - это выходные данные $l-1$ сверточного слоя, которые были дилатированы на 2^{l-1}.

Общая архитектура WaveNet представляет собой стек L сверточных слоев, каждый из которых имеет дилатацию, которая увеличивается в геометрической прогрессии. После каждых L слоев используется один слой $1x1$ свертки для сокращения размерности и сглаживания [2].

Наконец, на выходе WaveNet генерирует аудио-сигнал y_t, который может быть определен как:

$$y_t = \sum_{i=1}^{N} s_i \cdot softmax(c_i)$$

где N - это количество возможных значений аудио-сигнала (обычно 256 для 8-битных сигналов), s_i - это i-ое возможное значение аудио-сигнала, c_i - это выходной вектор последнего сверточного слоя, соответствующий i-ому значению аудио-сигнала.

Таким образом, WaveNet может быть описан как генеративная модель,

которая принимает на вход случайный шум и генерирует последовательность аудио-сигналов [3].

Обучение WaveNet производится путем минимизации функции потерь, которая определяется как отрицательная логарифмическая функция правдоподобия:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(y_t|y_{<t})$$

где T - это длина аудио-сигнала, y_t - это t элемент аудио-сигнала, а $y_{<t}$ - это все элементы аудио-сигнала до t элемента. Функция правдоподобия $P(y_t|y_{<t})$ определяется как:

$$P(y_t|y_{<t}) = softmax\big(f(y_{<t})\big)_{y_t}$$

где f - это функция, определенная нейронной сетью WaveNet.

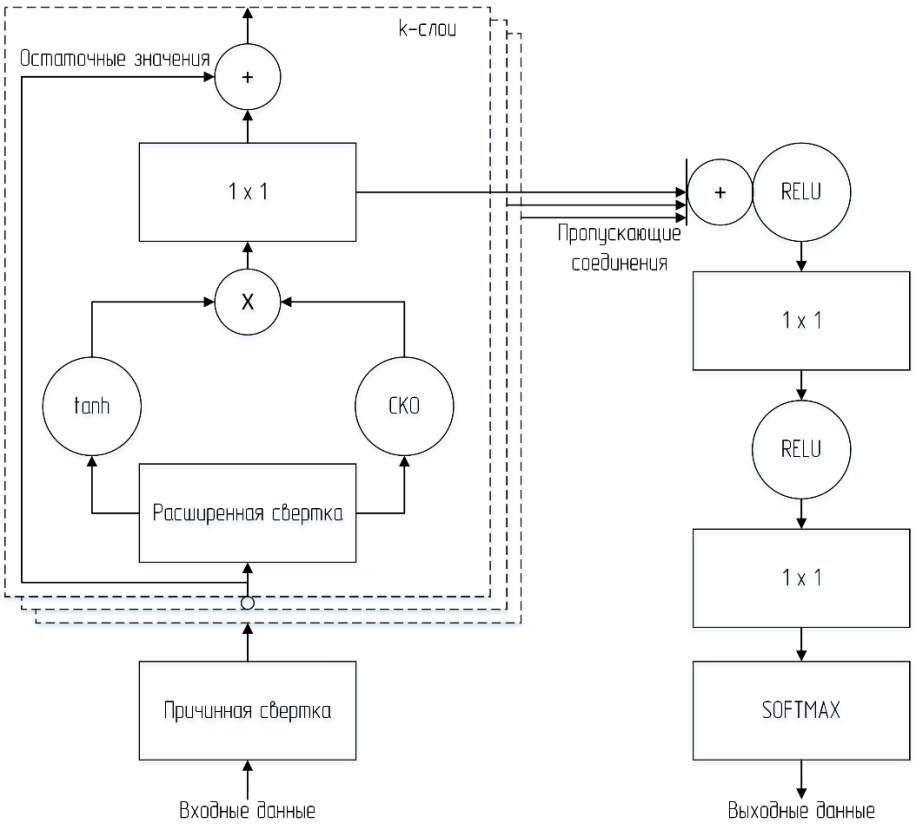

Рис. 1 – Архитектура оригинальной модели WaveNet

На рисунке 1 представлена полная архитектура оригинальной модели WaveNet. Стоит отметить две очень важные детали: резидуальные соединения и

скип-соединения. Каждый сверточный слой в WaveNet имеет резидуальное соединение (residual connection), которое добавляет к выходу слоя входные данные. Это позволяет бороться с проблемой затухания градиентов в глубоких нейронных сетях и упрощает процесс обучения. Каждые несколько сверточных слоев в WaveNet имеют скип-соединения (skip-connection), которые позволяют обойти несколько слоев и передать информацию на следующий слой. Это помогает сети более эффективно извлекать признаки из длинных последовательностей [4].

В целом, WaveNet представляет собой глубокую сверточную нейронную сеть, которая работает с последовательностями аудио-сигналов. Она использует сверточные слои с длиной ядра 2 и резидуальные и скип-соединения, чтобы извлекать признаки из последовательности и генерировать новые аудио-сигналы.

Модифицированная архитектура WaveRNN. WaveRNN - это модификация WaveNet, которая использует рекуррентные нейронные сети (RNN) вместо сверточных слоев для генерации аудио-сигналов. Основная идея WaveRNN заключается в использовании RNN для управления генерацией последовательности аудио-сигналов, где каждый элемент последовательности зависит от предыдущих элементов.

Для того чтобы улучшить эффективность генерации аудио-сигнала, в WaveRNN применяется метод иерархической генерации. Этот метод позволяет снизить количество вычислений, необходимых для генерации аудио-сигнала, за счет того, что модель не генерирует аудио-сигнал целиком, а разбивает его на некоторое количество уровней (layer). На каждом уровне модель генерирует более низкочастотные компоненты аудио-сигнала, которые затем используются на более высоких уровнях для генерации более высокочастотных компонент [5].

Для каждого уровня WaveRNN использует отдельную ячейку GRU. Входным сигналом для каждой ячейки является конкатенация выходов всех ячеек на более низких уровнях, что позволяет учесть все предыдущие компоненты аудио-сигнала при генерации текущего компонента. Выход каждой

ячейки используется для генерации компоненты на текущем уровне.

На высшем уровне WaveRNN генерирует самую высокочастотную компоненту аудио-сигнала, которая затем конкатенируется с компонентами на более низких уровнях для получения полного аудио-сигнала.

WaveRNN с использованием иерархической генерации позволяет достичь качества генерации аудио-сигнала, сопоставимого с WaveNet, при существенно меньшем количестве вычислений и параметров модели.

Основным отличием WaveRNN от WaveNet является использование рекуррентных нейронных сетей (RNN) в качестве генератора, вместо сверточных (Conv) блоков в WaveNet. Это позволяет уменьшить количество параметров и сократить время обучения [6].

Кроме того, WaveRNN использует иерархическую генерацию, когда аудиосигнал разбивается на несколько масштабов, каждый из которых генерируется отдельной моделью. Это позволяет увеличить детализацию генерируемого звука и уменьшить время генерации.

Процесс генерации аудио сигнала в WaveRNN можно разбить на несколько шагов:

1. Инициализация скрытого состояния: На первом шаге инициализируется скрытое состояние h_0.

2. Генерация скрытого состояния: На каждом шаге модель генерирует скрытое состояние h_t, используя предыдущее скрытое состояние h_{t-1} и текущий входной вектор x_t.

3. Генерация распределений вероятностей: Используя скрытое состояние h_t, модель генерирует распределения вероятностей $P(c_t)$ и $P(f_t)$ для текущего шага.

4. Выбор элементов из распределений: Для генерации отдельных элементов аудио сигнала модель выбирает соответствующие элементы из распределений $P(c_t)$ и $P(f_t)$.

5. Обновление скрытого состояния: После генерации элемента аудио сигнала на текущем шаге, скрытое состояние h_t обновляется, учитывая векторы

управляющей вентиляции u_t, вентиляции сброса r_t и вектор памяти e_t.

6. Повторение процесса: Шаги 2-5 повторяются для каждого элемента аудио сигнала, пока не будет сгенерирован весь сигнал.

Таким образом основная задача модели WaveRNN является обновление скрытого состояния на каждом шаге генерации аудио сигнала [7].

Задача WaveRNN – обрабатывать данные от генератора текста в речь (мел-спектрограммы) и звуковые данные говорящего и синтезировать речь на основе этих данных, этот процесс изображен на рисунке 2.

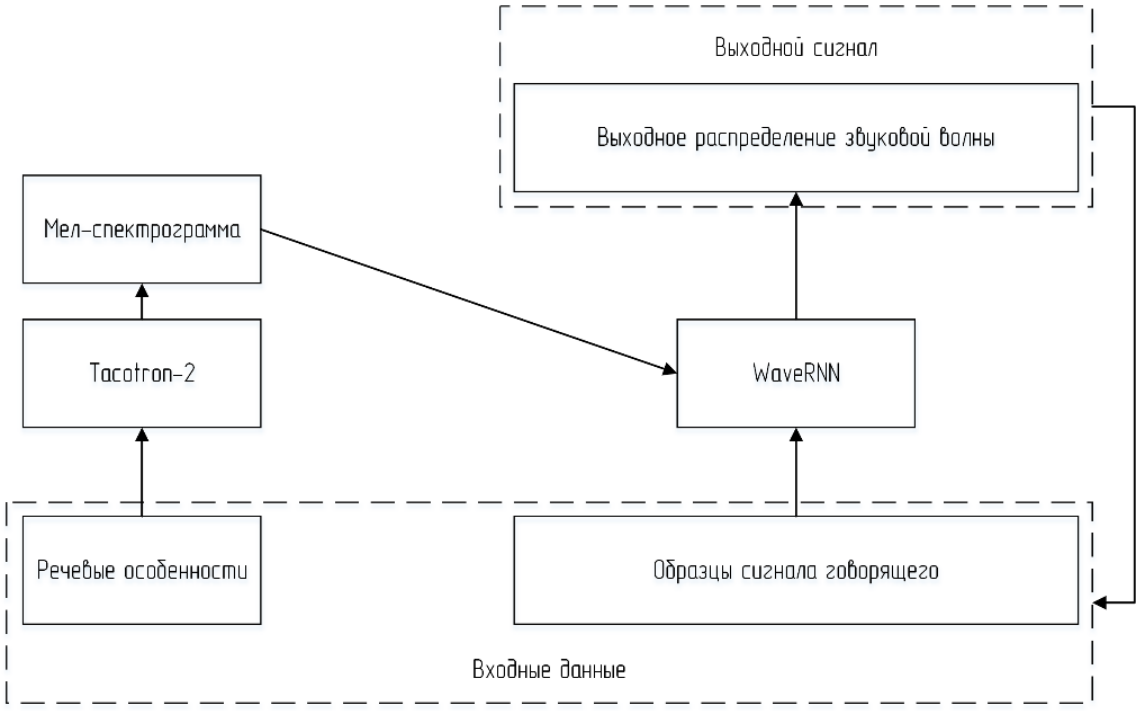

Рис. 2 – Основные процессы в работе TTS-системы на основе WaveRNN

Таким образом WaveRNN можно использовать в качестве вокодера в Text-to-Speech (TTS) системе, и конечная архитектура системы будет выглядеть следующим образом:

1. Преобразование текста в представление с использованием глубоких нейронных сетей (например, Tacotron 2). Tacotron 2 генерирует представление речи, называемое мел-спектрограмма.

2. Преобразование мел-спектрограммы в условное представление c_t, которое используется в WaveRNN в качестве входного вектора на каждом шаге

генерации.

3. Генерация аудио сигнала с помощью WaveRNN. На каждом шаге генерации WaveRNN использует условное представление c_t, сгенерированное на шаге 2, чтобы синтезировать аудио сигнал.

4. Обработка аудио сигнала для улучшения качества звука (например, с использованием WaveGlow, что является еще одним вариантом генеративной модели синтеза звука) [8].

Выводы. В данной работе рассматривается RNN модификация модели WaveNet для её использования в реализации вокодере в системе преобразования текста в речь на основе голоса говорящего. WaveRNN является более эффективной моделью по сравнению с WaveNet. В то время как WaveNet требует значительных вычислительных ресурсов и времени для обучения и генерации аудио, WaveRNN может достичь сопоставимого качества звука с более низким количеством параметров и вычислительной сложностью. WaveRNN способен генерировать аудио в режиме реального времени, что означает, что он может создавать звук непосредственно по мере поступления входных данных. Это полезно для приложений, требующих быстрой генерации речи, например, виртуальные ассистенты или системы распознавания речи. В целом, выбор вокодера для TTS-системы является важным шагом, который должен быть основан на компромиссе между качеством, скоростью и ресурсами, доступными для разработки и эксплуатации системы.

Библиографический список:

1. Белоножко П.Е., Белов Ю.С. Модификации архитектуры WaveNet для реализации вокодера в генеративной модели преобразования текста в речь // Научное обозрение. Технические науки. 2022. № 6. с. 37-42.

2. Мосин Е. Д., Белов Ю.С. Генерация музыки с использованием двунаправленной рекуррентной нейронной сети // Научное обозрение. Технические науки. 2023. № 1. с. 10-14.

3. H. Kanagawa, Y. Ijima. Multi-Sample Subband Wavernn Via Multivariate

Gaussian // ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2022. pp. 8427-8431.

4. J. M. Valin, J. Skoglund. LPCNET: Improving Neural Speech Synthesis through Linear Prediction // ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019. pp. 5891-5895.

5. J. Nistal, S. Lattner, G. Richard. Comparing Representations for Audio Synthesis Using Generative Adversarial Networks // 2020 28th European Signal Processing Conference (EUSIPCO). 2021. pp. 161-165.

6. P. Verma, C. Chafe. A Generative Model for Raw Audio Using Transformer Architectures // 2021 24th International Conference on Digital Audio Effects (DAFx). 2021. pp. 230-237.

7. Shen J. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018. pp. 4779-4783.

8. Y. C. Wu, T. Hayashi, P. L. Tobing. Quasi-Periodic WaveNet: An Autoregressive Raw Waveform Generative Model With Pitch-Dependent Dilated Convolution Neural Network // IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2021. vol. 29, pp. 1134-1148.