Ресурси за лекция по МЛ в STT and TTS

Презентация:
https://docs.google.com/presentation/d/1UVCGtKpPiXKsORwxK4BOTQVxDazXQcwXvmXBRZ3EefQ/edit#slide=id.gd486b3fbff_0_171

Intro: На проблема
Speech to Text CTC - слайдове

Text to Speech
Basic structure на text to speech
Deep Voice Structure
WaveNet
Tacotron 2
Transformers TTS

Demo page Microsoft + Others

End to with Gergana

Подходи
Sequence to sequence networks
Classification and clusterization

SEQUENCE to Sequence
Бележки
Такотрон е първият сек2сек нетуърк. Гугъл го предлагат. Използва рнн и декодер, който създава мел спектрограма, от която се създава аудио изхода. Има подобрение, Такотрон2, който опростява архитектурата. Принципът и на двете версии на архитектурата е да генерира мел спектрограми от текст, а след това да ги използва, за да генерира аудиоформи чрез уейв нет.
DCTTS е CNN. Избрана е подобна архитектура, защото рнн са бавни за трениране. Подобрение като се използва по-добър атеншън механизъм, използвайки факта, че има директен мапинг между входа и изхода на мрежата. По-нататъчно подобрение може би е постигнато с използване на трансформери.

Пейпъри
deep voice:
http://media.speech.zone/images/Simon_King_Lancaster_2019_compressed_for_publication.pdf
WaveNet: https://arxiv.org/abs/1609.03499
Tacotron: https://arxiv.org/pdf/1703.10135.pdf
Tacotron 2: https://arxiv.org/abs/1712.05884v2

DCTTS: https://arxiv.org/abs/1710.08969
Deep Voice 3: https://arxiv.org/pdf/1710.07654.pdf
An improved Tacotron: https://arxiv.org/pdf/1903.07398v1.pdf
Transformers TTS: https://arxiv.org/abs/1809.08895

Технологии и датасетове
Мозила отворен ттс:
https://github.com/mozilla/TTS

Coqui, a startup providing open speech tech for everyone
https://github.com/coqui-ai

Претренирани Такотрон модели, плюс нужното инфо за трениране на свой:
https://github.com/keithito/tacotron/

Джовани слайдове

Notes Preslav:

Slide 1
Здравейте, Ние сме Преслав и Джовани и ще ви представим методи за Speech to Text и Text to Speech.

Всеки от вас има на телефона си Google Assistant, Siri, Cortana. Методи за разпознаване на текст от реч и обратното са влезли в Google Home, Alexa продуктите.

Slide 2
Продуктите за speech to text, работят по следния начин. Имаме аудио което се подава, От него се създава спектограма (вълнова диаграма на звука), която се разбива на малки части. Това става в акустичния компонент. След това от акустичните единици се разпознават фонеми, това става в фонетичния модул, модел на произнасяне(По определение фонемата е звук или група от звуци, за които говорещите съответния език смятат, че изпълняват една и съща функция. Пример за фонема е */k/* в думите ***к**ол* и ***к**ил*. Като транскрипцията на думите). От тези фонеми се създава в езиковия модел думи, фрази и тнт,

От гледна точка на невронните мрежи. Имаме поредица от входове x1 до xT, искаме тези входове да ни дадат изходи от текст y1 до yL. Като X е спектограмата и ние искаме да намерим шанса P(Y|X) за това входа е определен изход.

Но ако тръгнем със стандартни методи за класификацията/регресия да решаваме проблема ще стигнем до следните спънки:
Проблемите на пръв поглед за простите алгоритми за supervised learning

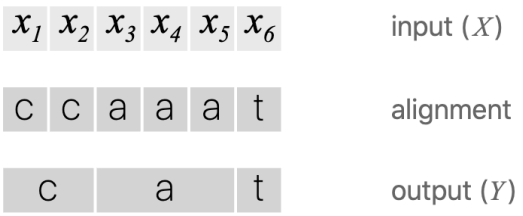
X и Y варират по дължина и по отношение.

Нямаме точно съответствие между X и Y - Предсказани символи too може да са от дума to Или too.

Slide 3
Затова и идва решението на проблема с Connectionist Temporal Classification

CTC алгоритъма преодолява тези проблеми. За дадено X ни дава като резултат разпределение от всички възможни Y. Може да използваме разпределението за да заключим какъв е резултата и да оценим шанса му.

Slide 4
Пример: input: x1, x2,x3,x4,x5,x6 alignment: ccaaat output: cat
CTC работи като сумира вероятността на всички възможни разпределения на символи. Трябва разберем какви са тези разпределения за да можем ще се решаваме loss функцията.



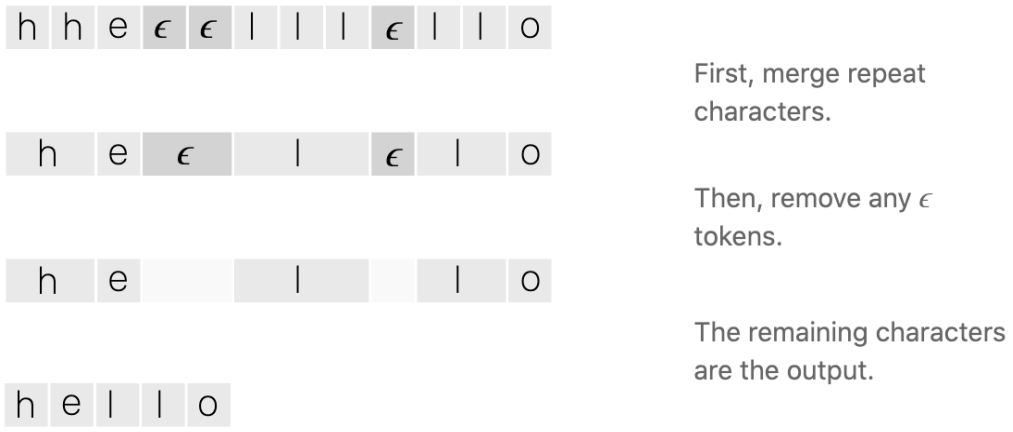
Интуитивен подход: Групираме по символи, всеки от тези символи стават нашия изход

Проблемите са два основни:
Не можем да имаме повтарящи се редици от символи.

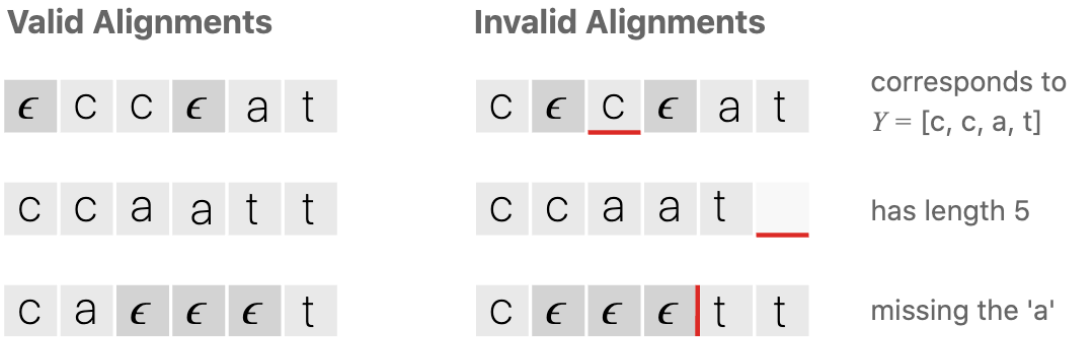
Не винаги input под формата на спектограма, трябва да произведе символ. Често имаме тишина и странични шумове.

Slide 5
За да реши тези проблеми CTC представя нов специален token и множество от позволени изходи. Token се нарича blank token. *Тук се използва е за него*. Той не отговаря за нищо и можем да го премахнем от изхода.

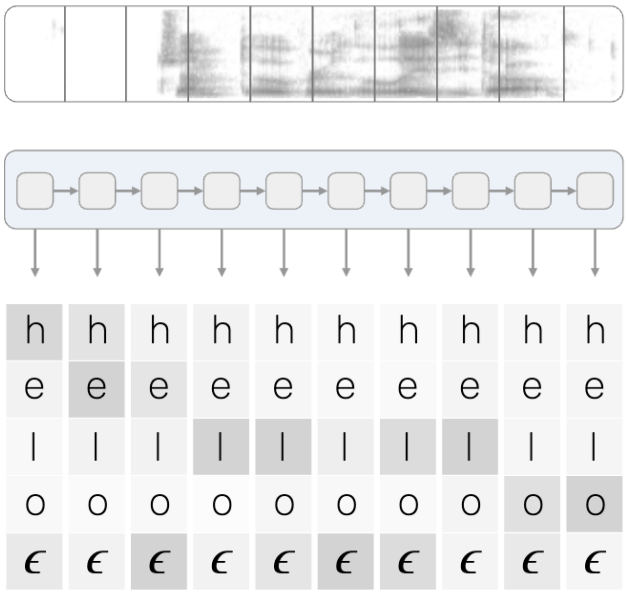
Така между повтарящи се символи имаме blank token. Така можем да ги групираме отделно след това махаме този blank token.



Slide 6



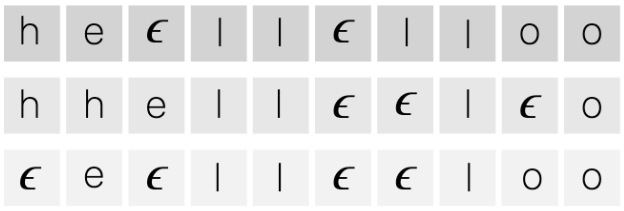
Slide 7



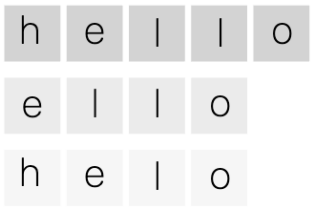
We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.



With the per time-step output distribution, we compute the probability of different sequences



By marginalizing over alignments, we get a distribution over outputs.

Сглобен алгоритъма работи както следва.

Подаваме спектрограма като вход

Отделните отрязъци влизат в RNN невронна мрежа.

За всеки отрязък се предсказва символ от нашата азбука в случая { h, e, l, o, blank}

Спрямо предсказаните символи изчисляваме възможните разпределения за изход.

Махаме дублираните символи и blank tokens

И спрямо резултата за вероятността за различните изходи, вземаме най големия и той ни създава крайния резултат.

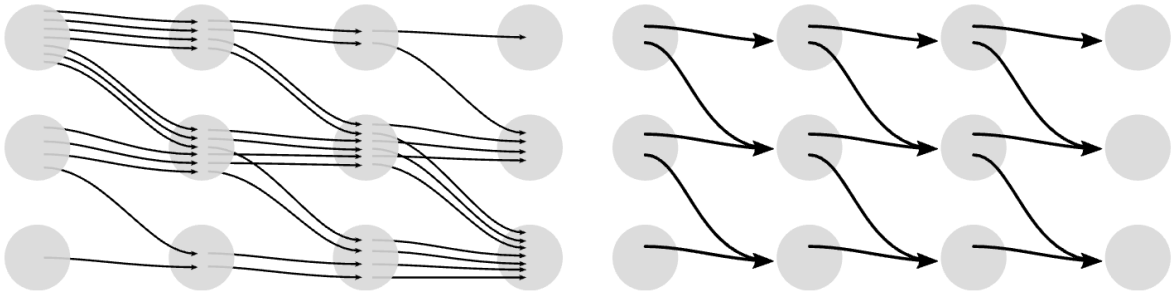
Вероятност за едно разпределение (Умножението на вероятността за всеки символ)

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

The CTC conditional **probability** | **marginalizes** over the set of valid alignments | computing the **probability** for a single alignment step-by-step.

Slide 10 - Виждаме че от всеки символ може да следва всеки следващ.
Както видяхме по горе, всеки неврон от мрежата ни дава разпределението от азбуката.

Slide 11
Изчислението на CTC loss функцията може да е много скъпо. Затова се налага да оптимизираме чрез динамично програмите. Ако на някоя стъпка две разпределения са стигнали до един и същ изход то можем да обединим (като фибоначи):



Summing over all alignments can be very expensive.

Dynamic programming merges alignments, so it's much faster.

Сега ефикасно можем да изчислим loss функцията, следваща стъпка е изчислим градиента и да трениране модела. CTC loss функцията се диференцира спрямо вероятността във дадено време. От там правим backpropagation както винаги.

$$\sum_{(X,Y)\in\mathcal{D}} -\log\ p(Y \mid X)$$

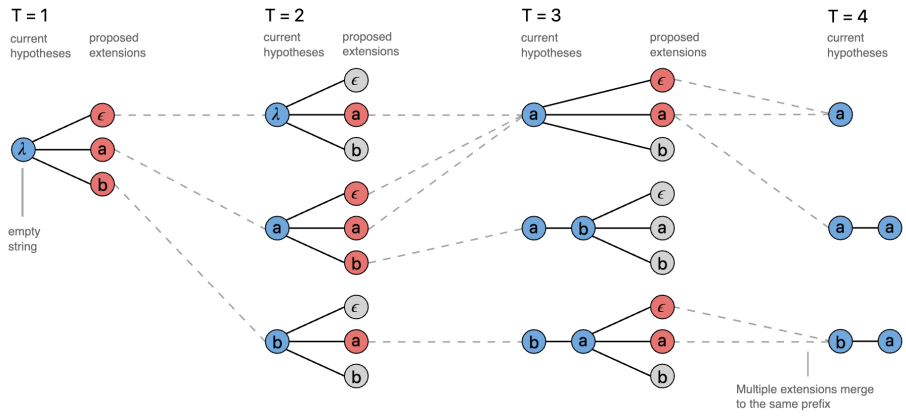
Slide 13

Правене на заключение:

Вариант 1:
Вземаме максималната вероятност на всяка стъпка.

$$A^* = \underset{A}{\operatorname{argmax}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

Вариант 2: Beam Search, който се свързва с повтарящите се символи.



Slide 14:

Минуси:

Независимост на възможните изходи.
Пример: не можем да предскажем че tripple A е AAA. Затова seq2seq са по добри обаче с езиков модел може да се преодолеят тези недостатъци.

Често прави правописни грешки или се обърква граматиката. Затова се интегрират езикови модели които да решат тези проблеми и да re-rank-нат пътищата.

Why not seq2seq

RNN е по бавно се обучава

https://distill.pub/2017/ctc/

Demo code:
https://github.com/apoorvnandan/speech-recognition-primer/blob/master/bare_bones_asr.py

Other Text:

Now what does Sequence-to-Sequence models have anything to do with Connectionist Temporal Classification(CTC)? Well, CTC is an algorithm used to tackle a key issue faced when training seq-2-seq models, which is, when the length of the input and output sequences do not match.

https://www.clics-network.org/downloads/comparison-decoding-strategies.pdf

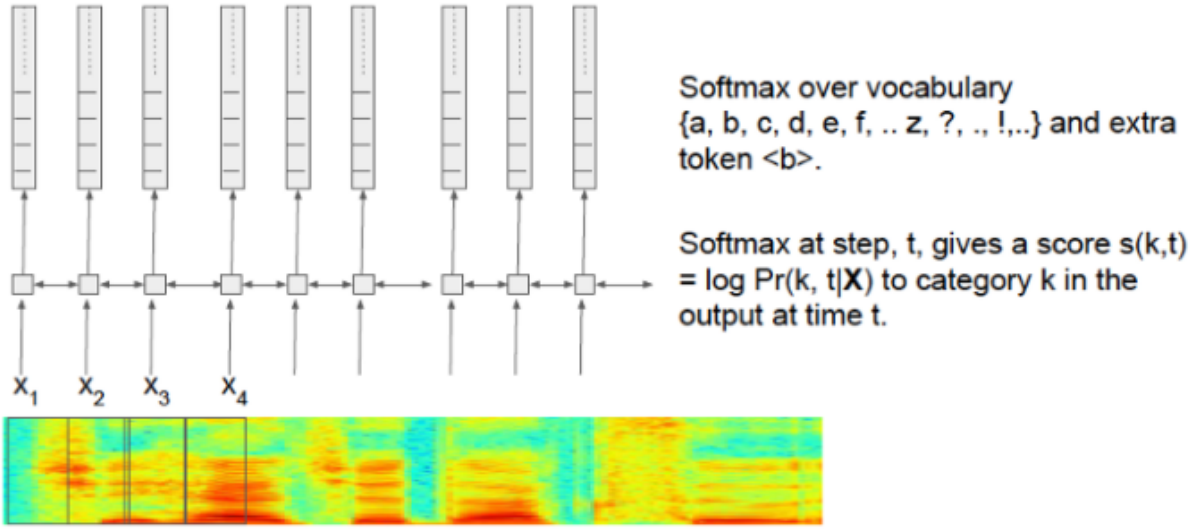
The spectrogram input can be thought of as a vector at each timestamp. A 1D convolutional layer extracts features out of each of these vectors to give you a sequence of feature vectors for the LSTM layer to process.

https://towardsdatascience.com/hello-world-in-speech-recognition-b2f43b6c5871

Connectionist Temporal Classification (CTC). X е поредица от части на спектограма с размер T(от Time): x1, x2, …, xT, и резултати Y L (от Language): y1, y2, …, yL. T > L.

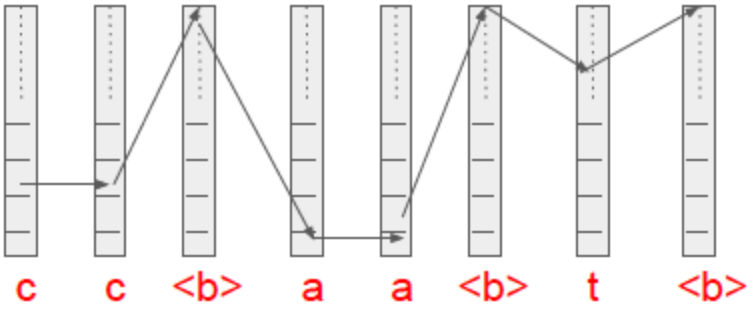
- Получаваме като input spectrogram. Тя се зарежда bi-directional RNN.
- В края имаме softmax за всеки отрязък от време. Softmax function се прилага до определна абзука. Тук азбуката е малките букви на английския език. Ще имаме и допълнителен символ за CTC - blank token.

- Всеки резултат от предсказването е вероятността за определен token в това време. Резуктата е score s(k, t) - log вероятност за категория k във време t при данни X.



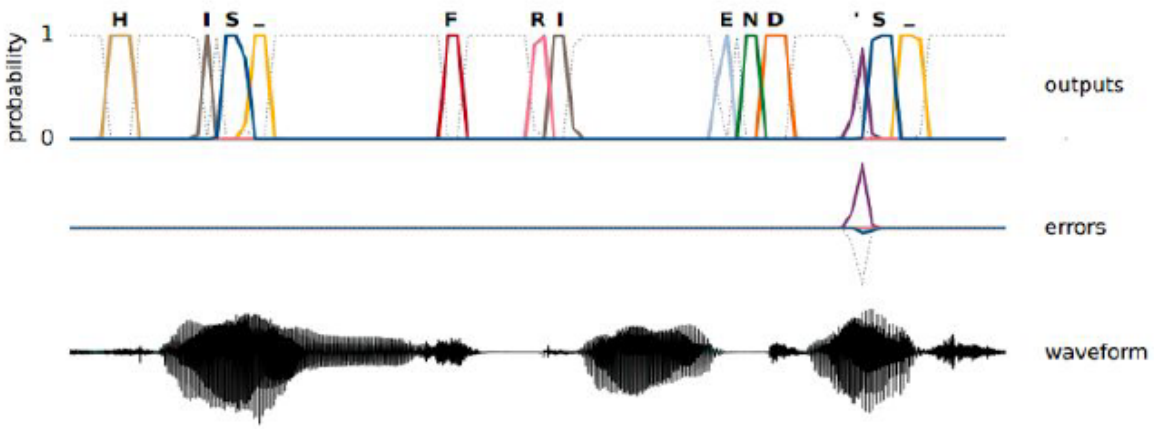
CTC модела може да репрезентира всички тези пътища в пространството от softmax функции и да гледа символите които отговаря за това време

От снимката виждаме CC blank AA blank T blank.



Ако проследим пътищата с ограничение че може да се движим само през същата фонема от една стъпка към друга. Така се получават различни пътища за представяне на крайната редица. Може да имаме **cc <b> aa <b> t <b>** , **cc <b> <b> a <b> t <b>** о, **cccc <b> aaaa <b> tttt <b>.** Така излиза че имаме експоненциално много възможности които да доведат до крайния резултат и създаваме тежък за решаване от време проблем. Но с помощта на динамично програмиране можем да оптимизираме пътищата (като фибоначи) и да изчислим вероятността и градиента. Този градиент може да бъде back propagated до невронната мрежа, чийто параметри да оптимизираме.

Снимка как изглежда предсказването.



Model learns to make peaky predictions!

Пише че това е модела за *OK Google*.

Speech to Text

Deep Voice
The system comprises five major building blocks:
a segmentation model for locating phoneme boundaries,
a grapheme-top-honeme conversion model,
a phoneme duration prediction model,
fundamental frequency prediction model,
an audio synthesis model.

Wave Net

WaveNet is speech synthesis network by Google. It is only part of the pipeline, but it was first of its kind. It generates audio including speech based n the previous output. The sampling happens with frequency of 16khz to 24khz. This is too fast for rnn-s. It is fast for cnn-s as well. Cnn-s could be improved by dilation, allowing for sampling from thousands of steps in the past.

Tacotron 2

The system is composed of a recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel-scale spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time-domain waveforms from those spectrograms.

TTS with Transformers

Although end-to-end neural text-to-speech (TTS) methods (such as Tacotron2) are proposed and achieve state-of-the-art performance, they still suffer from two problems:
1) low efficiency during training and inference;
2) hard to model long dependency using current recurrent neural networks (RNNs).
That's why some improvements have been demonstrated with the help of transformers. The use of only multi-head attention mechanism allows for dropping recurency and parallelization of training. Also, any two inputs at any time are connected by a self-attention mechanism, which solves the long dependency problem.

Demo page Microsoft
https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/

“Text to Speech е Speech to Text Случайност не мисля
”