[course site]

#DLUPC

Day 4 Lecture 1

Text-to-Speech

Antonio Bonafonte
antonio.bonafonte@upc.edu

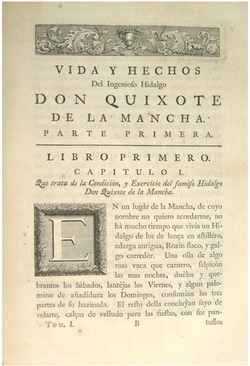
Associate Professor
Universitat Politècnica de Catalunya

Table of contents

Introduction to Text-to-Speech (TTS)

TTS: Text-to-Speech

```
tts.say("You have 3 new messages.")
```



VIDA Y HECHOS

Del Ingenioso Hidalgo

DON QUIXOTE

DE LA MANCHA.

PARTE PRIMERA.

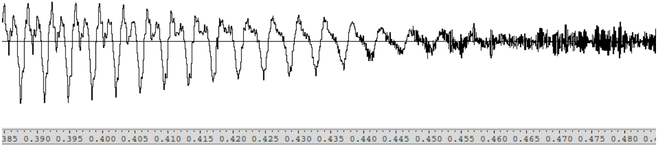
LIBRO PRIMERO.

CAPITULO I.

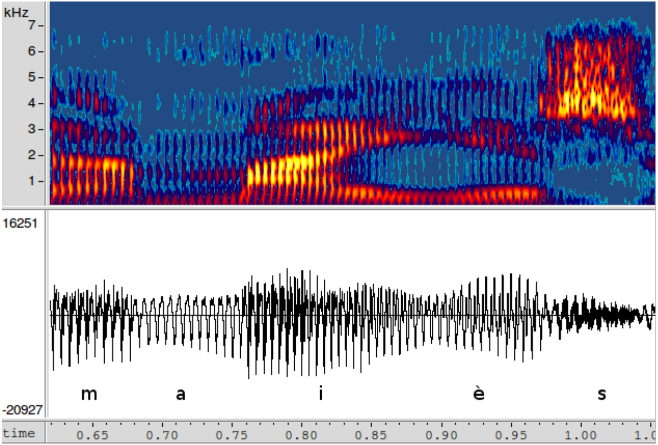
Que trata de la Condiciòn, y Exercicio del famòso Hidalgo Don Quixote de la Mancha.

EN un lugàr de la Mancha, de cuyo nombre no quiero acordarme, no hà mucho tiempo que vivìa un Hidalgo de los de lança en astillèro, adarga antìgua, Rozin flaco, y galgo corredòr. Una olla de algo mas vaca que carnero, salpicòn las mas noches, duèlos y quebrantos los Sàbados, lantèjas los Viernes, y algun palomino de añadidura los Domingos, consumìan las tres partes de su hazienda. El resto della concluỳan sayo de velarte, calças de velludo para las fiestas, con sus pantuflos

TOM. I. B

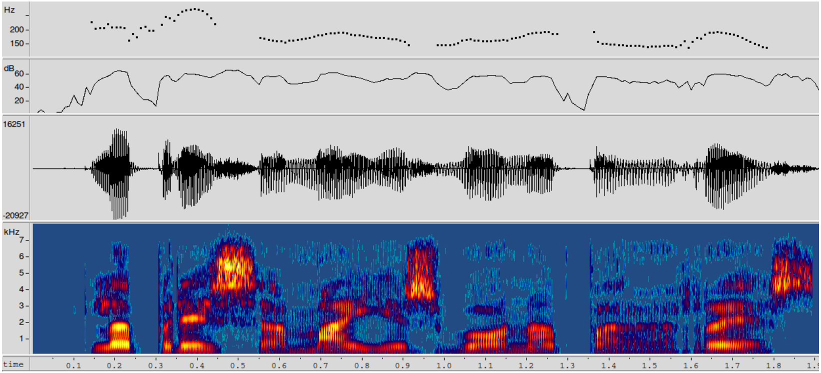

385 0.390 0.395 0.400 0.405 0.410 0.415 0.420 0.425 0.430 0.435 0.440 0.445 0.450 0.455 0.460 0.465 0.470 0.475 0.480

10 0ms



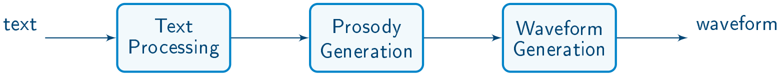
kHz
7
6
5
4
3
2
1
16251
-20927
m
a
i
è
s
time
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00

400 ms



$\approx$ 2 seconds

Classic TTS diagram



1. Text Processing

- Text normalization
- PoS tagger
- Phonetic Transcription

1.1. Text normalization I

Transform input text into fully expanded words

Case review

- Numbers (ordinals, cardinals, dates, times, telephone, IDs, IP, codes, roman numbers, maths, numbers in brands,
- Addresses, Abbreviations, Acronyms
- Punctuation characters
- SMS & *internet* writtings (typos, smileys, etc.)

1.1. Text normalization II

Some examples on the cover of recent newspaper (14/12/2017):

`1-O` `21-D`

`ERC` `PSOE` `Cs` `PP` `CUP`

`ITV` `TSJC` `sentencia del TC` `SEAT GTI` `CCCB`

`Al Jazira, 1 - R. Madrid, 2`

`80` `OT 2017` `57%`

`22:25 Polònia`

`EUR/USD`

Kaggle Challenge 2017: Google Text Normalization

1.2. PoS Tagger I

What gets us into trouble is not what we don't know.
It's what we know for sure that just ain't so.

Mark Twain

What	**gets**	**us**	**into**	**trouble**	**is**	**not**	**what**	**we**	**do**	**not**	**know**	**.**
what	get	us	into	trouble	be	not	what	we	do	not	know	.
WP	*VBZ*	*PRP*	*IN*	*NN*	*VBZ*	*RB*	*WP*	*PRP*	*VBP*	*RB*	*VB*	*Fp*

It	**'s**	**what**	**we**	**know**	**for**	**sure**	**that**	**just**	**ai**	**not**	**so**	**.**
it	be	what	we	know	for	sure	that	just	be	not	so	.
PRP	*VBZ*	*WP*	*PRP*	*VBP*	*IN*	*JJ*	*IN*	*RB*	*VBP*	*RB*	*RB*	*Fp*

From FreeLing

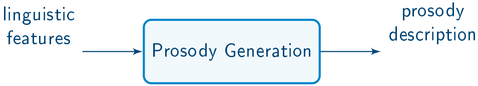
1.3. Phonetic Transcription

Transcribe words into symbolic phonetic representation (IPA, SAM-PA, etc.)

1. Look up into phonetic lexicon (CMU, Unisyn, LC-STAR, etc.)
2. Letter-to-sound *rules* for unknown words.
   Classification Trees (CART), HMM, Finite-state transducers (FST), Pronunciation-by-Analogy (PbA), Neural Networks (NN)

Example

what	gets	us	in	trouble
w 'V t	g 'E t_h s	@ s	I n	t_h r 'V b l



Prosody:

- Indicate end of sentences and their structure.
- Semantic disambiguate.
- Emphasize words or phrases.
- Express emotion, affect, position with respect to verbal information,

Only some cues are included in the text → invent.

- Phrasing (breaks)
- Prominence
- Intonation
- Timing

Speakers group words while speaking.

Example

In terms of money [B] he was no better off.
John and Sara [B] were running very quickly.
John and Sara were running [B] very quickly.

Basic Features

Punctuation
PoS in window around the word
Distance in words/syllables to previous/next punctuation/break

Classification or transcription

word_features $\rightarrow$ (B | $\neg$B)
HMM or Finite state transducers (FST)
Level Building
Recurrent Neural Networks

Prominence, Emphasis, Accent ...

Some words receive extra strength compared with neighbor.

`Parking Lot, City Hall`
`It was John that pick up your call.`
`The profit was $10 M before tax, not after tax`
`A German teacher vs.  a German teacher`

Similar features and classification methods than phrasing.

There are different representations of prosody, ranging from
Symbolic/Phonologic Representation, E.g.: ToBI. to
pure Acoustic Representation, e.g., log $F0$ contour.

Observed movements

Intonational patterns in prosodic phrases
Declination during prosodic phrases
Characteristic endings (., ?, Wh-questions, continue)
Accents

Features

Punctuation of the prosodic phrase

Duration of phrase (#words, #syl) and sentence

Phrase/Word/Syllable/Phoneme position

PoS, content vs function word

Prominency, lexical stress

Representation

Unit: phrase vs. accent group vs. superpositional vs. syllable vs. phoneme

Symbolic (e.g. ToBI indexes and tones)

Acoustic: parametric log $F0$ (k values, polynomial coef., etc.)

Phonemes (or syllables) duration

Observed movements

Intrinsic phoneme duration
Lengthening in last syllable of prosodic phrase
Lengthening in prominent words and stressed syllable
Reduced in function words
Influence of neighbors phonemes

Modeling of phonemes (or syllable)

Similar features than for intonation
Regression models (CART, Neural Networks, ...)

Duration of pauses is also very important. Features used for break detection are used to estimate pause duration.

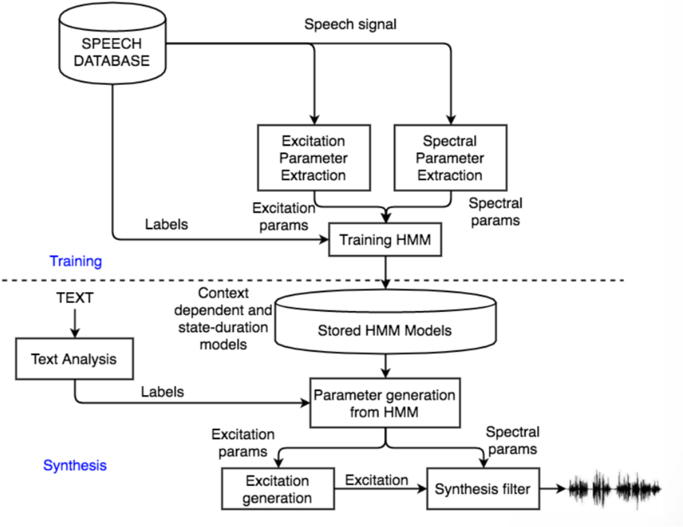
3. Waveform Generation

Approaches

	Articulatory and Physical Models
80s	Rule based synthesis (e.g. Klatt's formant synth.)
90s	Synthesis by Concatenation
1996–	+ corpus based unit selection
1996–	Statistical/Parametric synthesis (SPSS)
2010–	Deep Learning

Statistical Parametric Speech Synthesis

Statistical Parametric Speech Synthesis (SPSS)



Speech features

In SSPS the signal is represented by *parameters*

Need to be invertible and with reduced quality loss.

Typically: source/filter model $\rightarrow$ spectral envelope and excitation (F0, etc.)

- Vocoder: LPC Vocoder, MELP, Straight, Vocaine, World
- For each sliding window (shift $\approx$ 5ms) compute:
  - Spectral envelope. Goal: high resolution, stable
  - Pitch, F0
  - Rich Excitation. E.g.: voiced band representations

Linguistic features

Spectral envelope: correlated with phoneme, context phonemes (coarticulation), emphasis, maybe type of word

Duration: correlated with phoneme, sentence length, sentence structure, stress, break position, ...

F0: same as duration

Excitation detail: F0 and phoneme, context, speech rate ...

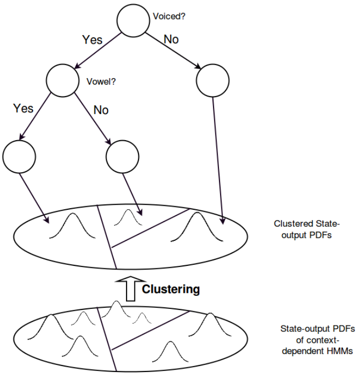
```
_^_-pau+b=w~1_1/A:0_0_0/B:0-0-1~1-1&0-1#0-0$0-0!0-0;0-1|_/C:0+1+3/D:SIL_0/E:SIL+1~0+1&0+0#0+0/F:AQ_2/G:0_0/H:0=0~2=1|0/I:4_3/J:5+6-1
_^pau-b+w=e~1_3/A:0_0_1/B:0-1-3~1-2&1-4#0-0$0-2!0-0;0-2|e/C:0+0+3/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
pau^b-w+e=n~2_2/A:0_0_1/B:0-1-3~1-2&1-4#0-0$0-2!0-0;0-2|e/C:0+0+3/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
b^w-e+n=o~3_1/A:0_0_1/B:0-1-3~1-2&1-4#0-0$0-2!0-0;0-2|e/C:0+0+3/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
w^e-n+o=z~1_3/A:0_1_3/B:0-0-3~2-1&2-3#0-0$1-2!0-0;1-1|o/C:0+1+2/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
e^n-o+z=D~2_2/A:0_1_3/B:0-0-3~2-1&2-3#0-0$1-2!0-0;1-1|o/C:0+1+2/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
n^o-z+D=a~3_1/A:0_1_3/B:0-0-3~2-1&2-3#0-0$1-2!0-0;1-1|o/C:0+1+2/D:SIL_0/E:AQ+2~1+3&0+3#0+1/F:UNKNOWN_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
o^z-D+a=a~1_2/A:0_0_3/B:0-1-2~1-1&3-2#0-0$1-1!0-0;2-1|a/C:0+1+2/D:AQ_2/E:UNKNOWN+1~2+2&1+2#1+1/F:NC_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
z^D-a+a=s~2_1/A:0_0_3/B:0-1-2~1-1&3-2#0-0$1-1!0-0;2-1|a/C:0+1+2/D:AQ_2/E:UNKNOWN+1~2+2&1+2#1+1/F:NC_1/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
D^a-a+s=pau~1_2/A:0_1_2/B:0-1-2~1-1&4-1#0-0$2-0!0-0;1-0|a/C:0+0+0/D:UNKNOWN_1/E:NC+1~3+1&2+1#1+0/F:SIL_0/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
a^a-s+pau=_~2_1/A:0_1_2/B:0-1-2~1-1&4-1#0-0$2-0!0-0;1-0|a/C:0+0+0/D:UNKNOWN_1/E:NC+1~3+1&2+1#1+0/F:SIL_0/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
a^s-pau+_=_~1_1/A:0_1_2/B:0-0-1~1-1&0-5#0-0$0-3!0-0;1-0|_/C:0+0+0/D:NC_1/E:SIL+1~0+4&0+3#1+0/F:SIL_0/G:0_0/H:4=3~2=1|0/I:0_0/J:5+6-1
```

Example of linguistic information (HTS representation)

- One acoustic model for each phoneme → one label for each phoneme
  - Phoneme context: identity; type: plosive,affricate,...
  - Syllable context
  - Word context
  - Sentence context

Training

The number of contexts (labels) is too big to be train with any speech database
$\rightarrow$ clustering, typically using decision trees



Parameter generation

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{X}|\mathcal{L}) \approx \operatorname{argmax}\{p(\mathbf{q}|\mathcal{L}) \cdot p(\mathbf{X}|\mathbf{q}, \mathcal{L})\}$$

To simplify equation, $\mathbf{q}$ and $\mathbf{X}$ are derived in two steps.

$$\hat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmax}}\, p(\mathbf{q}|\mathcal{L})$$

$$\hat{\mathbf{X}} \approx \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{X}|\hat{\mathbf{q}}, \mathcal{L})\}$$

First equation: duration prediction ($\mathbf{q}$ is the state sequence)
Second equation: acoustic generation

Duration Model

Model the duration (#frames at the state) using pdf.
E.g.:$f(d) = \mathcal{N}(d|\mu, \sigma)$

Training

- After the acoustic models have been estimated, compute how many frames are used at each state.
- Learn the mapping function between linguistic features and pdf parameters (e.g., decission tree).

Inference

- Get the pdf parameters given the linguistic features.
- Use z to adjust speaking rate: $d = \mu + z \cdot \sigma$
  ($z = 0$, same speaking rate as training)

Direct solution $\rightarrow$ unnatural speech.

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{X}|\hat{\mathbf{q}}, \mathcal{L})\} = (\mu_{q_1}, \mu_{q_2}, \dots, \mu_{q_T})$$

No changes between consecutive frames. Unnatural dynamics. E.g.:
$\hat{\mathbf{X}} = (\boxed{\mu_{s_1}, \mu_{s_1}, \mu_{s_1}}, \boxed{\mu_{s_2}, \mu_{s_2}, \mu_{s_2}, \mu_{s_2}}, \dots)$

Solution: include in the model dynamic features: $\Delta\mathbf{X}, \Delta^2\mathbf{X}, \dots$

Acoustic Generation II

1. Dynamic features are related with static features
   E.g. $\Delta x_t = \frac{1}{2}(x_{t+1} - x_{t-1})$

$$[\mathbf{X}, \Delta\mathbf{X}] = \mathbf{W} \cdot \mathbf{X}$$

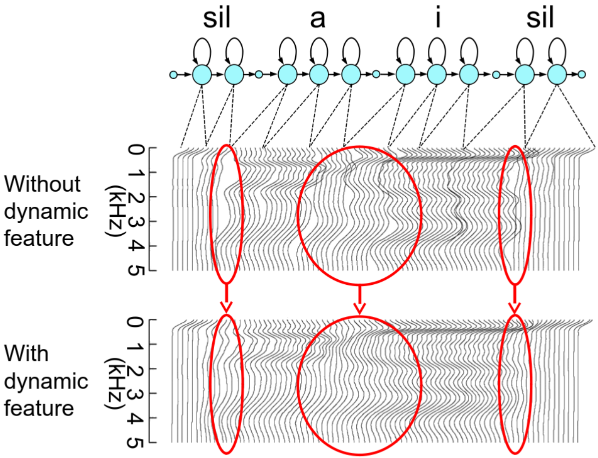
2. Model:

$$p(\mathbf{X}, \Delta\mathbf{X} | \mathcal{L}) = p(\mathbf{W} \cdot \mathbf{X} | \mathcal{L})$$

3. Generation:

$$\hat{\mathbf{X}} = \operatorname*{argmax}_{\mathbf{X}} p(\mathbf{W} \cdot \mathbf{X} | \hat{\mathbf{q}}, \mathcal{L})\}$$

Acoustic Generation III



From HTS.Group 2015

Flexibility

1. Statistical Parametric Speech Synthesis is robust with respect to training data (e.g. noise, pronunciation errors).
2. Is it possible to use adaptation techniques to create new voices (speakers, styles) adapting the parameters of existing voice.
3. It is also possible to interpolate voices.
   $\lambda = 0.25 \cdot \lambda_{\text{neutral}} + 0.75 \cdot \lambda_{\text{happy}}$
4. Averages reduces speech dynamics: some approaches to improve it. Post-filtering, global variance, trajectory models
5. Vocoder limits the quality: $\rightarrow$ high quality vocoders as straight, ahocoder, world

See also ...

I recommend the excellent tutorial:

Fundamentals and recent advances in HMM-based speech synthesis (HTS.Group 2015)

It includes audio samples with statics vs dynamics features, style and speaker adaptation, interpolation, global variance, straight vocoder ....

Synthesis by Concatenation

Synthesis by Concatenation

Alias Cut & Paste

Development

1. Define *synthesis units* (e.g., phones, *diphones)*)
2. Record all possible synthesis units.
3. Segment and add to the *unit database*

Operative

1. Get the linguistic and prosodic features of the input text.
2. Map phonemes to synthesis units.
3. Get the needed units from the database.
4. Concatenate *and play*.

Challenges

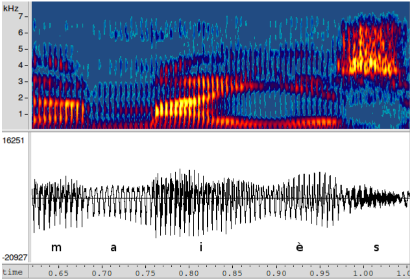
Prosody requirements

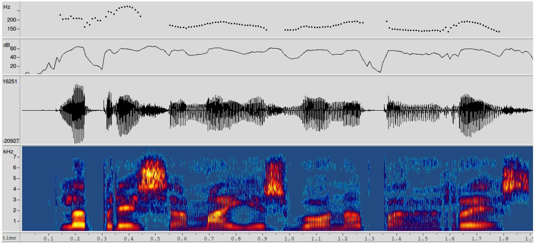
The synthesis units have to match the prosody requirements: e.g., segmental duration and F0.

Continuity requirements

Concatenation of units should avoid discontinuity in:

1. Waveform (zero crossing, interpolation)
2. Pitch
3. Phase
4. Frequency (diphones better than phones)


kHz
7
6
5
4
3
2
1
16251
-20927
m
a
i
è
s
time
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00


Hz
200
150
dB
60
40
20
16251
-20927
kHz
7
6
5
4
3
2
1
time
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8

Approach 1: Signal Processing

Record units in *neutral* phonetic context and prosody.

Apply signal processing methods to modify duration and F0 of speech and to smooth discontinuities.

- LPC: Vocoder using Linear Prediction Coding
- TD-PSOLA: time-domain pitch-synchronous overlap and add
- HSM: Harmonic & stochastic model
- ...

	Original	D x 0.7	D x 1.3	F0 +400 cents	F0 -400 cents
TDPSOLA	▶	▶	▶	▶	▶

Unfortunately, speech processing introduces distortion. Specially for large changes.

Approach 2: Corpus Based Unit Selection

1. Record several instances of each speech synthesis unit.
2. Select the *best ones*.

The recorded units should cover as much as possible different phonetic and prosodic context: sentences, designed corpus.

Larger databases $\rightarrow$ lesser signal processing

Typically: 1h – 15h, professional speaker, recording studio.

Automatic segmentation of synthesis units (HMM + Viterbi)

Selection Criterion I

Define the cost function and select units with smallest cost:

$$\hat{u}_1^N = \underset{u_1^N}{\operatorname{argmin}}\, C(t_1^N, u_1^N)$$

Goal: select the units:

- that match the *target* prosody
- and have good *concatenation* between consecutive units.

$$\hat{u}_1^N = \underset{u_1^N}{\operatorname{argmin}}\, \rho \cdot C^t(t_1^N, u_1^N) + (1-\rho) \cdot C^c(u_1^N)$$

C^t	target cost.
C^c	concatenation cost
ρ	target/concatenation compromise

Selection Criterion II

Typically, the cost functions are additive:

$$\hat{u}_1^N = \operatorname*{argmin}_{u_1^N} \rho \cdot \sum_{n=1}^{N} C^t(t_n, u_n) + (1-\rho) \cdot \sum_{n=1}^{N-1} C^c(u_n, u_{n+1})$$

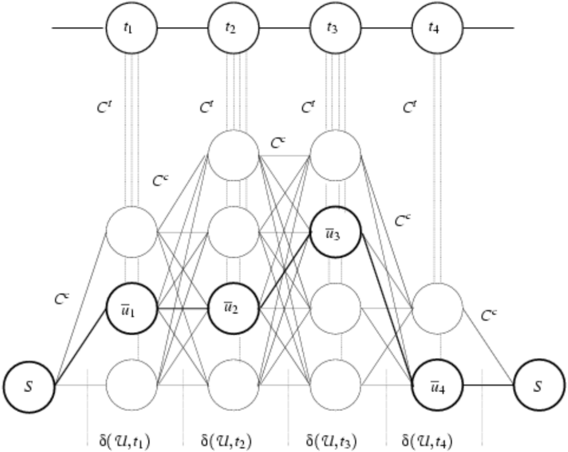
C^t: target cost. Measures match between target and unit.
C^c: concatenation cost. Measures discontinuity between end of one unit and beginning of next one.

Selection Criterion III

Target and Concatenation cost are composed by several *subcosts*.

- Each unit is characterized by linguistic context and acoustic features: $\{L, A\} = \{l_1, \ldots, l_R, a_1, \ldots, a_S\}$
- Each target is characterized by linguistic context and acoustic prediction: $\{L, \bar{A}\} = \{l_1, \ldots, l_R, \bar{a}_1, \ldots, \bar{a}_S\}$
- For each feature, target/concatenation subcosts are defined.

Search



Viterbi algorithm is applied to find the best units.

Deep Learning for Speech Generation

Deep Learning for Speech Generation I

DL in statistic parametric Speech Synthesis(Ling et al. 2015)

- Sustitute HMM means by DNN predictions (use same generation algorithm).
- Use RNN to directly generate smooth trajectories
- Interpolation and adaptation methods have been proposed(Pascual de la Puente 2016)

DL in Concatenative / Unit Selection

- Use the trajectories or embedded features in the target cost.
- Use MDN to predict pdf of acoustic features which are then used to asses the units
  Apple' Siri:(Capes et al. 2017)
  - Predict acoustic feature pdf ($f_\theta(y)$) from linguistic features. ($\theta = \phi(L)$)
  - Use the log $f_\theta(.)$ to asses the units.
  - Target cost: phoneme duration, F0 (beg. mid., and end)
  - Concatenation cost: ΔMFCC and ΔF0 in unit boundaries

Deep Learning for Speech Generation III

Towards end-to-end speech synthesis

Goal: from characters to samples.

Examples:

- Linguistic processing is being change by character reader (char2wav, tacotron).
- Vocoder waveform generation sustituted by neural vocoder (samplernn, wavenet).
- Some end-to-end system (integrated learning) (deepvoice).

Results are starting to be better than dominant unit concatenation techniques.

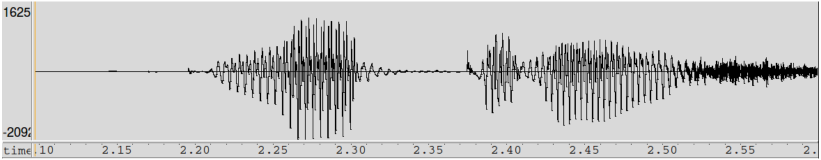
Wavenet: A Generative Model for Raw Audio (Oord et al. 2016

)

From *PixelCNN* $\rightarrow$ *Wavenet*

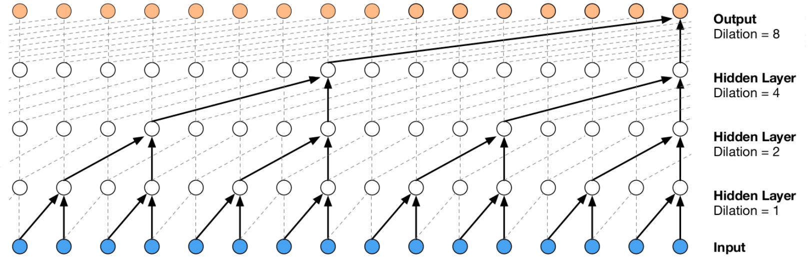
- Models the generation of waveform sample x_n, based on previous samples.
- Autoregressive & probabilistic model: $p_\theta(x_n|x_0, x_1, \ldots, x_{n-1})$
- $p_\theta(.)$ modeled using causal convolutional networks.
- Discrete values (8 bits, μ-law) $\rightarrow$ classification loss

Wavenet: dilated convolutions I



- Long dependencies need to be captured in speech & audio.
  - 100 ms at 16 kHz $\rightarrow$ 1600 samples
  - Difficult to capture for RNN; computational prohibitive for(conventional) CNN

Wavenet: dilated convolutions II



Dilated Convolution (from Oord et al. 2016)

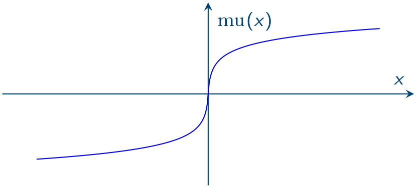
Wavenet: dilated convolutions III

- Dilated convolutions skip input values.
- Output has the same size than input.
- Stacked dilated convolutions: large *receptive field* with few layers and filter width.
- Dilation doubled and then repeated (stacked blocks).
  - Block of dilated convolutional networks with exponential dilated factor.
    Eg: $1, 2, 4, \ldots, 512 \rightarrow$ receptive field 1024 samples.
  - Four blocks stacked: receptive field 4096 samples.
  - If sampling frequency 16 000 kHz $\rightarrow$ receptive field: 256 ms.

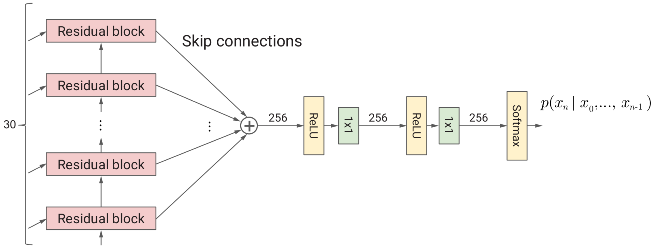
Softmax Distributions

- The authors claim that `softmax` distributions work better than MDN, even with continuous data.
- Apply μ-law, quantize to 8 bits $\rightarrow$ 256 *classes*.
- Cross-entropy loss.

$$\mathrm{mu}(x) = \mathsf{sign}(x)\,\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}$$



Architecture: skip connections

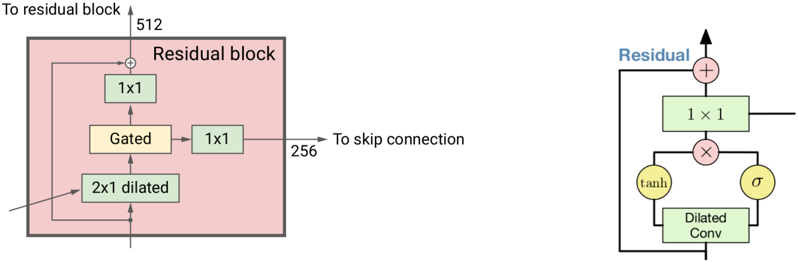


Architecture (from Zen 2017)

If I understand correctly ...

- There are 30 dilated convolutions ($\approx 9 \times 4$)
- Residual blocks output 256 feature maps
- *Parameterised* skip connections $\rightarrow$ speed up & deeper models

Architecture: gated activation units & residual connections



Residual block (from Zen 2017; Oord et al. 2016)

- Gated activation unit: $z = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x})$

Conditional Wavenets

- As it is, wavenet can produce *speech like* signals and invent piano music.
- Can be conditioned with speaker (multispeaker training) and with linguistic features (TTS).

Global conditioning h (e.g.: speaker)

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

Local conditioning h_t (e.g.: linguistic features)

$z = \tanh(W_{f,k} * \mathbf{x} + V_{f,k}^T * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k}^T * \mathbf{y})$

$\mathbf{y} = f(\mathbf{h})$: learned upsampling.

$V * \mathbf{y}$: 1×1 convolution.

TTS: Results

Speech samples	Subjective 5-scale MOS in naturalness	
	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

(from Oord et al. 2016)

1ex

Samples:

`https://deepmind.com/blog/wavenet-generative-model-raw-audio/`

Wavenet: application

Wavenet: a very good vocoder

Wavenet learn the a priori distribution of speech, unsupervised.

Can be conditioned to be combined with several systems:

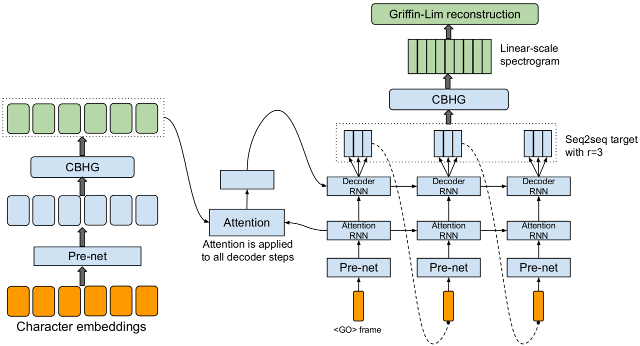
- Tacrotron 2 (end-to-end TTS)
- Low rate speech coding
- To *teach* other network (Wavenet 2)
- Speech enhancement
- etc.

Tacotron: Towards End-to-End Speech Synthesis (Wang et al. 2017)

From *PixelCNN* $\rightarrow$ *Wavenet*

- Synthesizes speech directly from text
  - No linguistic features: the input units are the characters of *normalized text*.
  - Output is the raw spectrogram (phase is recovered using the Griffin-Lim algorithm).
- Seq2seq with attention.
- Character represented using a elaborated architecture inspired in character-level neural machine translation (Lee, Cho, and Hofmann 2016).

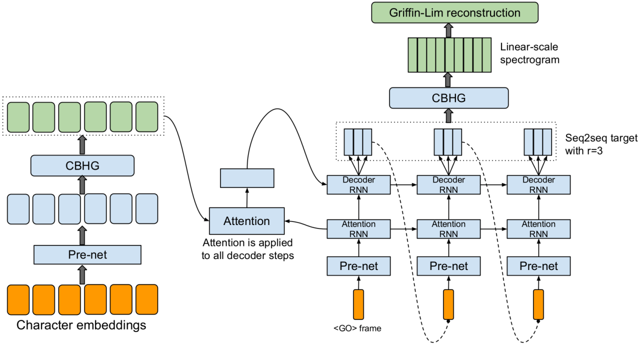
Architecture



Tacotron Architecture (from Wang et al. 2017)

seq2seq input: character

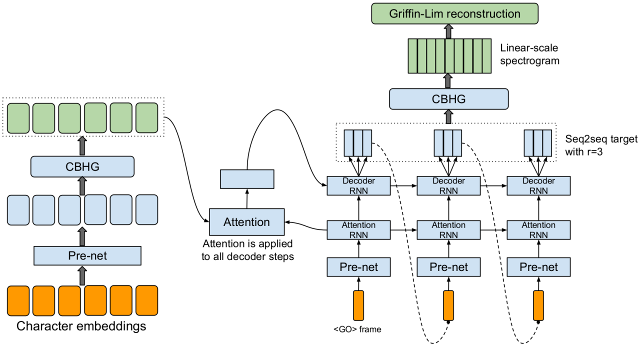
Architecture



Tacotron Architecture (from Wang et al. 2017)

seq2seq output: *r targets* (80-band mel spectrogram)

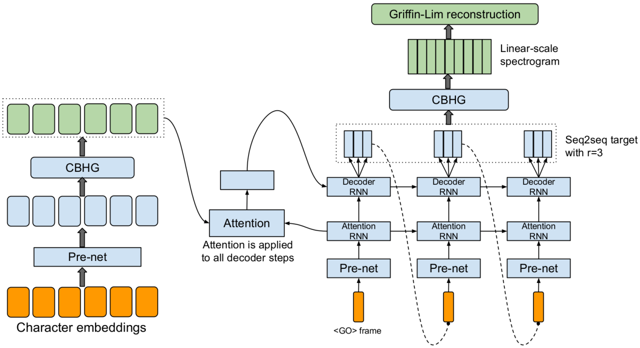
Architecture



Tacotron Architecture (from Wang et al. 2017)

Pre-net: Fully connected networks (2)

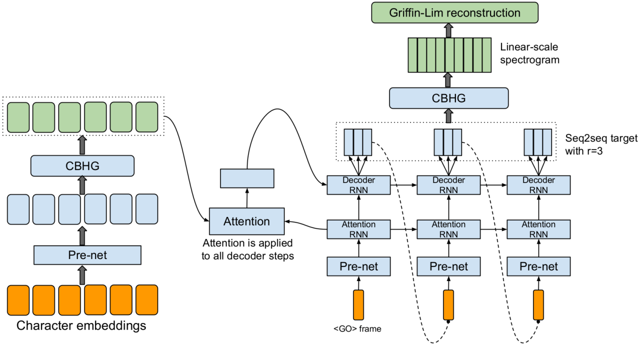
Architecture



Tacotron Architecture (from Wang et al. 2017)

CBHG: *Basically* 1D convolutional network + bidirectional GRU

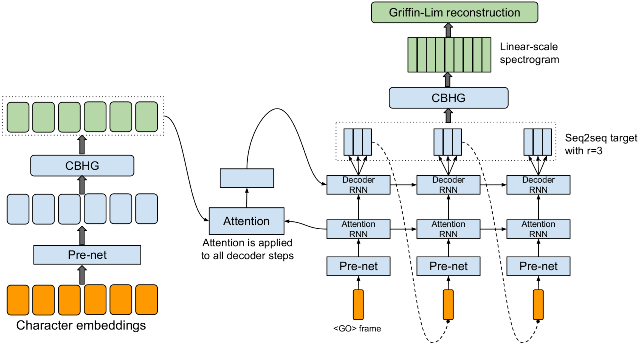
Architecture



Tacotron Architecture (from Wang et al. 2017)

Encoder CBHG: Character representation.

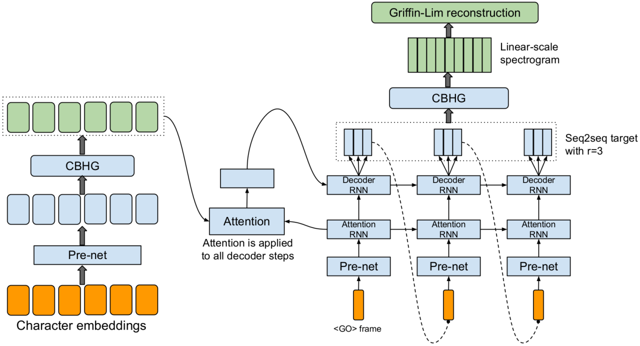
Architecture



Tacotron Architecture (from Wang et al. 2017)

Input to decoder: *transformed* previous target & context vector

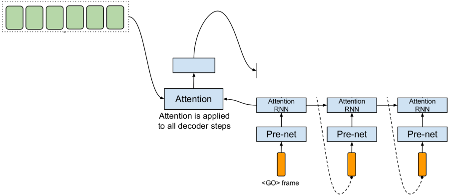
Architecture



Tacotron Architecture (from Wang et al. 2017)

Postprocessing CBHG: Transform targets in spectrogram.

Context vector c_t



Given K characters, the encoder CBGH produces K vectors, $\mathbf{e_k}$

For each t, the attention RNN predicts the coefficients $\alpha_t(k)$

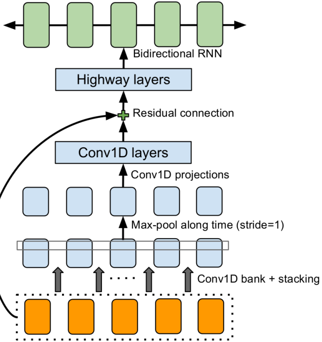
$$\mathbf{c_t} = \sum_{k=1}^{K} \alpha_t(k) \cdot \mathbf{e_k}$$

($\hat{\mathbf{y}}_{\mathbf{t-1}}$ and attention memory) $\rightarrow \alpha_{\mathbf{t}} \rightarrow \mathbf{c_t}$

($\hat{\mathbf{y}}_{\mathbf{t-1}}$, $\mathbf{c_t}$ and decoder memory) $\rightarrow \mathbf{y_t}(1:r)$

CBHG: Character representation

Caracter embedings (256), Prenet (128)
Convolution 1D: filter of differents widths: 1,2, ..., 16
Max pooling, width=2, stride=1
2 Conv 1D projection layers
4 Highway layers (FC+ ReLU)
Bidirectional GRU (128 units)



CHBG (from Wang et al. 2017)

CBHG post-processing: mel energies → spectrogram.

Tacotron Results

System	MOS
Tacotron	3.82
Parametric	3.69
Concatenative	4.09

5-Scale **M**ean **O**pinion **S**core

Samples:

`https://google.github.io/tacotron/`

Char2Wav: Reader

Encoder/decoder with attention

Encoder: bidirectional RNN

Input: chars or phonemes

Decoder: RNN with attention

Output: vocoder features

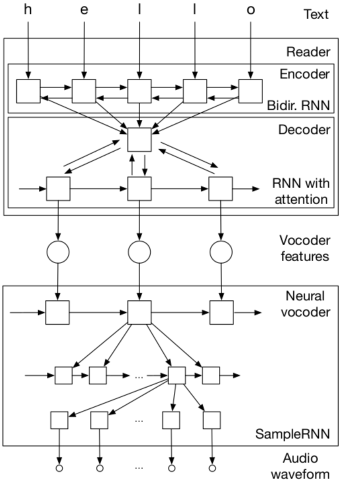
SampleRNN

Neural vocoder

Condition *SampleRNN* with vocoder features

Hierarchical structure to capture the different scales in audio

Char2Wav (Sotelo et al. 2017)



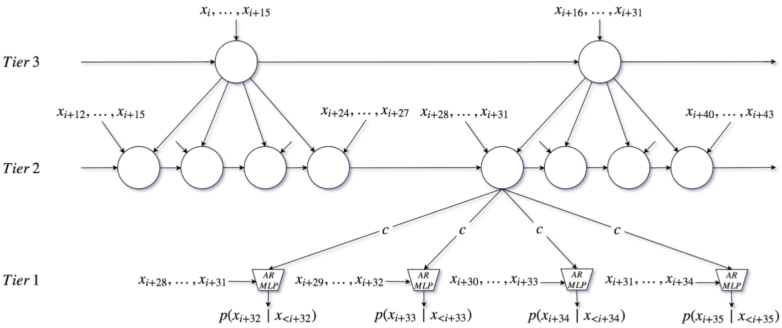
$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1}, h)$$

$$g_i = \sum_{j=1}^{L} \alpha_{i,j} h_j$$

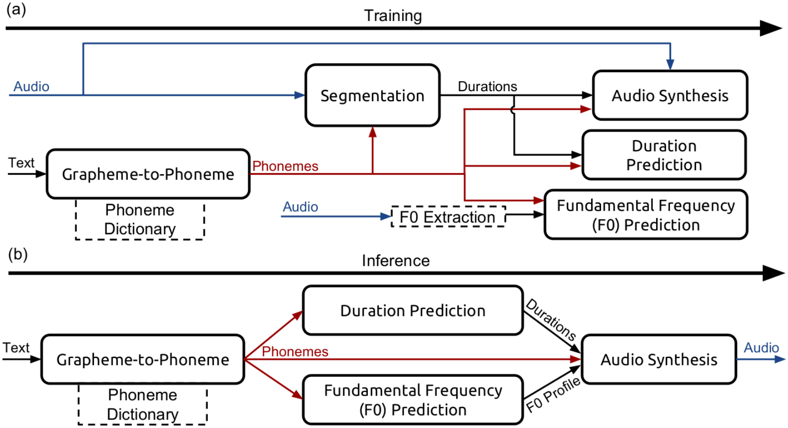
$$y_i \sim Generate(s_{i-1}, g_i)$$

$$s_i = RNN(s_{i-1}, g_i, y_i)$$

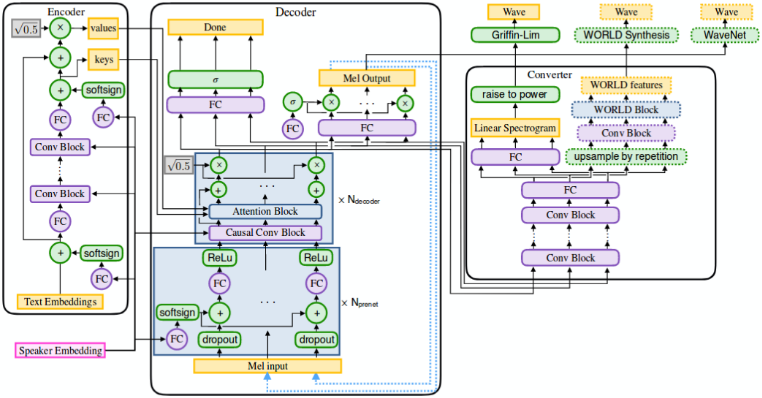
SampleRNN (Mehri et al. 2016)



DeepVoice (2017)



DeepVoice 3



Encoder
√0.5
values
keys
softsign
FC
Conv Block
Conv Block
FC
softsign
FC
Text Embeddings
Speaker Embedding
Decoder
Done
σ
FC
Mel Output
σ
FC
FC
√0.5
Attention Block
Causal Conv Block
× N_decoder
ReLu
ReLu
FC
FC
softsign
FC
dropout
dropout
× N_prenet
Mel input
Wave
Griffin-Lim
Wave
WORLD Synthesis
Wave
WaveNet
Converter
raise to power
Linear Spectrogram
FC
WORLD features
WORLD Block
Conv Block
upsample by repetition
FC
Conv Block
Conv Block

References

References

Capes, Tim et al. (2017). "Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System". In: *Proc. Interspeech 2017*, pp. 4011–4015. DOI: 10.21437/Interspeech.2017-1798. URL: http://dx.doi.org/10.21437/Interspeech.2017-1798.

Lee, Jason, Kyunghyun Cho, and Thomas Hofmann (2016). "Fully Character-Level Neural Machine Translation without Explicit Segmentation". In: *CoRR* abs/1610.03017. URL: http://arxiv.org/abs/1610.03017.

Ling, Z. H. et al. (2015). "Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends". In: *IEEE Signal Processing Magazine* 32.3, pp. 35–52. ISSN: 1053-5888. DOI: 10.1109/MSP.2014.2359987.

Mehri, Soroush et al. (2016). "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model". In: *CoRR* abs/1612.07837. URL: http://arxiv.org/abs/1612.07837.

Oord, Aaron van den et al. (2016). "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*. URL: https://arxiv.org/abs/1609.03499.

References II

Pascual de la Puente, Santiago (2016). "Deep learning applied to speech synthesis". MA thesis. Universitat Politècnica de Catalunya. URL: https://github.com/santi-pdp/msc_thesis.

Sotelo, Jose et al. (2017). "Char2Wav: End-to-end speech synthesis". In:

Wang, Yuxuan et al. (2017). "Tacotron: Towards End-to-End Speech Synthesis". In: URL: https://arxiv.org/abs/1703.10135.

Zen, Heiga (2017). *Generative Model-Based Text-to-Speech Synthesis*. Invited talk given at CBMM workshop on speech representation, perception and recognition. URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45882.pdf.

Credit to figures

HTS.Group (2015). *Fundamentals and recent advances in HMM-based speech synthesis*. URL: hts.sp.nitech.ac.jp.