Yandex

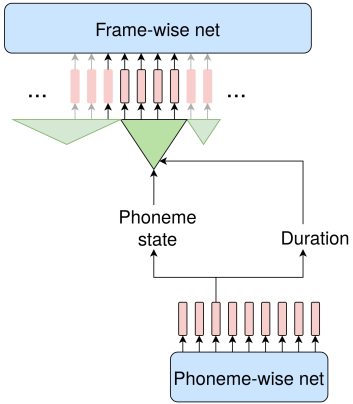
SCHOOL OF DATA ANALYSIS

TTS: Acoustic Models Extensions

Vladimir Kirichenko

May 16th 2022

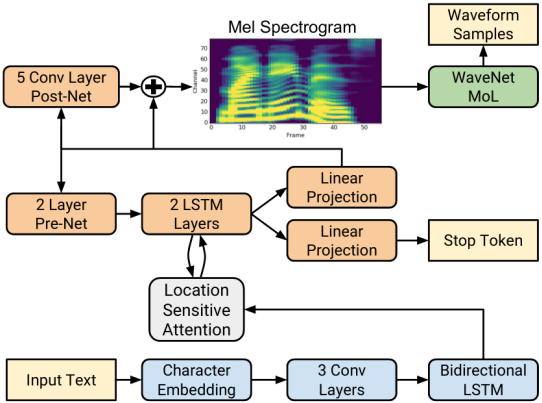
Recap: AMs Nowadays



- RNN + Upsampling

[1] Link Deep Voice 2

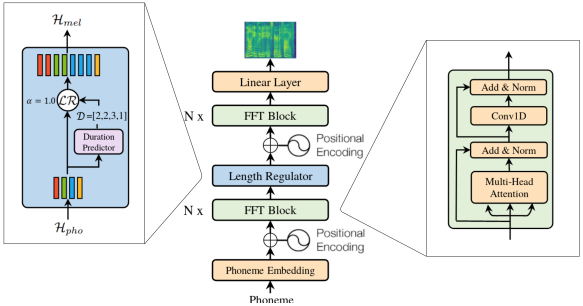
Recap: AMs Nowadays



- RNN + Upsampling
- RNN + (monotonness-aware) Attention

[1] Link Tacotron 2

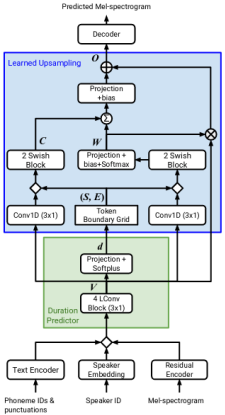
Recap: AMs Nowadays



- RNN + Upsampling
- RNN + (monotonness-aware) Attention
- FFT + Upsampling

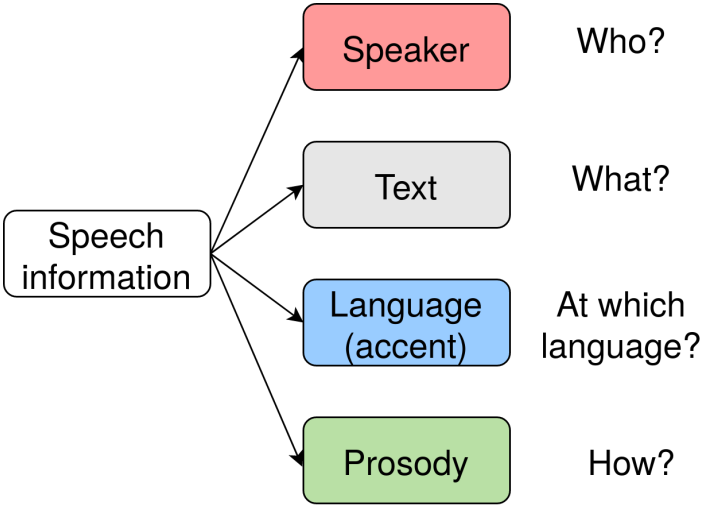
[1] Link Fast Speech

Recap: AMs Nowadays



- RNN + Upsampling
- RNN + (monotonness-aware) Attention
- FFT + Upsampling
- FFT + Soft-upsampling + Soft-DTW loss

[1] Link Parallel Tacotron 2


Speech information
Speaker
Who?
Text
What?
Language (accent)
At which language?
Prosody
How?

Speech Information

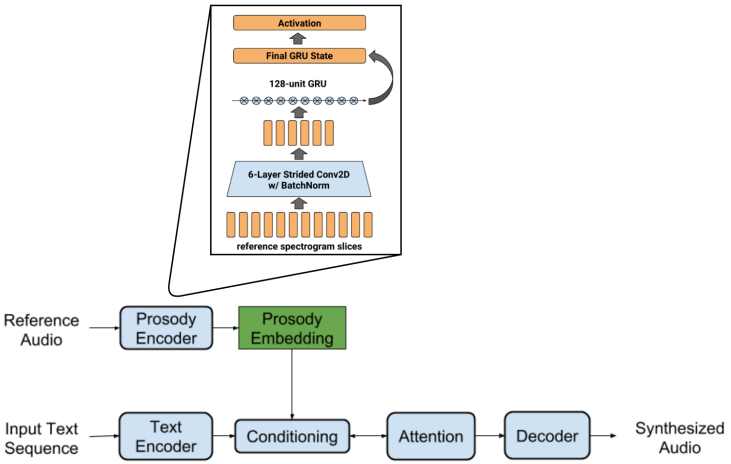
- Let's split the components! (explicitly or implicitly)
- We'll get **control** (style, accent, speaker parameters)
- We'll **use** the corpus **more effectively** (need less hours per single speaker)
- We'll **save training time** (one model to rule them all)

Speech Styles

Prosody

- Prosody - intonation, emotion, expression, temp
- Many Prosodies can correspond to a single text
- Model trained with MSE tends to over-smooth (mode collapse) without prosody information
- Model with random (dropout, SGD, VAE) tends to random the intonation without prosody information

Global Style Hint



Let's add hint:

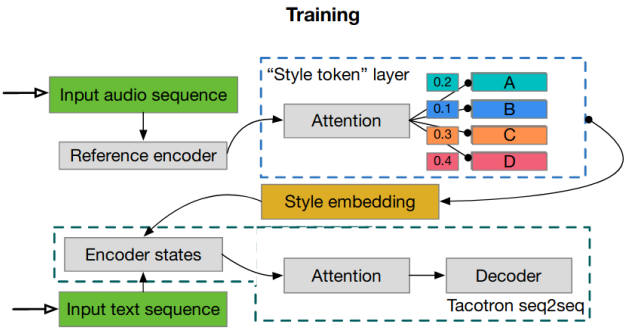
$$P = \text{AudioEncoder}(mel_{gt})$$
$$E = \text{TextEncoder}(text)$$
$$E^* = [E; P]$$

[1] Link Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron

Data Leak Problems

- With the hint we introduce leakage into training procedure
- We need to get these hints at the inference to generate proper audio
- With the AudioEncoder passing too much data we will be unable to do that
- We need Prosody Embeddings to be constant, predictable or able to be randomized from the input
- So, we need to restrain data leakage

Style Tokens



- Let prosody embedding be a weighted sum of a block of (constant) vectors — *style tokens*
- The only part depending on the audio — weights these tokens are summed with
- Attention is used to determine the weights (query is encoded prosody, keys are tokens)

[1] Link Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis

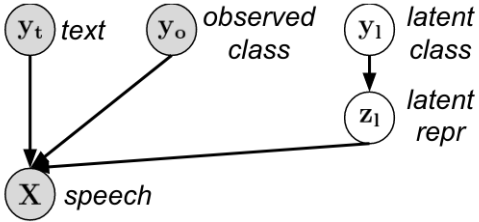
Inference With GST

- At the inference stage we can use hard-coded value for GST:
  - An pre-recorded utterance with good prosody
  - An utterance suitable for the context
  - Something similar to the phrase spoken in recordings set
- Or we can predict GST from the text encoder state:
  - By training predictor at a specific subset we can specify reading style
  - Or we can train GST predictor together with the net by adding a loss component:

$$\mathcal{L}_{GST} = \|\text{GST}(mel_{gt}) - \text{GSTPred}\,(\text{Encoder}(text))\|_2^2$$

[1] Link Predicting Expressive Speaking Style From Text In End-To-End Speech Synthesis

Probabilistic Approach



- $\mathbf{X}$ - audio
- $\mathbf{Y}_t$ - text
- $\mathbf{y}_o$ - known parameter (e.g. speaker ID, emotion etc.)
- $\mathbf{y}_l$ - GMM gaussian ID
- $\mathbf{z}_l$ - GMM gaussian mean

- Generation with AM can be represented as fitting a probabilistic model:

$$p(\mathbf{X}|\mathbf{Y}_t, \mathbf{y}_o)$$

- Let's introduce style information as some latent variable with GMM distribution - $(\mathbf{y}_l, \mathbf{z}_l)$
- Generation with style-aware AM turns into fitting a joint model:

$$p(\mathbf{X}, \mathbf{y}_l, \mathbf{z}_l|\mathbf{Y}_t, \mathbf{y}_o) =$$

$$p(\mathbf{X}|\mathbf{Y}_t, \mathbf{y}_o, \mathbf{z}_l)p(\mathbf{z}_l|\mathbf{y}_l)p(\mathbf{y}_l)$$

Probabilistic Approach

- Let q be the neural network that approximates the posterior: $q(\mathbf{y}_l|\mathbf{X})q(\mathbf{z}_l|\mathbf{X}) \approx p(\mathbf{y}_l, \mathbf{z}_l|\mathbf{X}, \mathbf{Y}_t, \mathbf{y}_o)$
- Then we can train the network like VAE, maximizing the score:

$$\text{LLH}(\mathbf{X}|\ldots) - \text{Dist}\left(q(\mathbf{y}_l, \mathbf{z}_l||p(\mathbf{y}_l, \mathbf{z}_l))\right)$$

- Which takes the form:

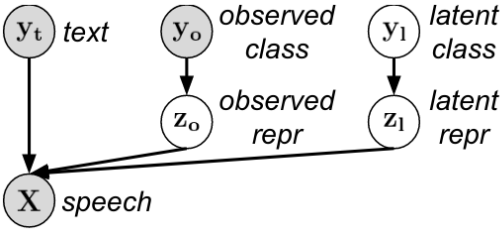
$$\mathbb{E}_{p(\mathbf{z}_l|X)}\left[\log p(\mathbf{X}|\mathbf{Y}_t, \mathbf{y}_o, \mathbf{z}_l)\right] - D_{KL}\left(q(\mathbf{y}_l, \mathbf{z}_l||p(\mathbf{y}_l, \mathbf{z}_l)\right) \approx$$

$$\mathbb{E}_{q(\mathbf{z}_l|\mathbf{X})}\left[\log p(\mathbf{X}|\mathbf{Y}_t, \mathbf{y}_o, \mathbf{z}_l)\right] - \mathbb{E}_{q(\mathbf{y}_l|X)}\left[D_{KL}\left(q(\mathbf{z}_l|\mathbf{X})||p(\mathbf{z}_l|\mathbf{y}_l)\right)\right] - D_{KL}\left(q(\mathbf{y}_l|\mathbf{X})||p(\mathbf{y}_l)\right)$$

- We can estimate $q(\cdot)$ via sampling

[1] Link Hierarchical Generative Modeling for Controllable Speech Synthesis

Incorporating Known Data



- The same way we can deal with known parameters $\mathbf{y}_o$
- Let's introduce a gaussian for each known class
- Let z_o be a center for it and let $q(z_o|X)$ be a subnet estimating it

The loss will be:

$$\mathbb{E}_{q(\mathbf{z}_o|\mathbf{X})q(\mathbf{z}_l|\mathbf{X})}\left[\log p(\mathbf{X}|\mathbf{Y}_t, \mathbf{z}_o, \mathbf{z}_l)\right]$$
$$-D_{KL}\left(q(\mathbf{z}_o|\mathbf{X})||p(\mathbf{z}_o|\mathbf{y}_o)\right)$$
$$-\mathbb{E}_{q(\mathbf{y}_l|X)}\left[D_{KL}\left(q(\mathbf{z}_l|\mathbf{X})||p(\mathbf{z}_l|\mathbf{y}_l)\right)\right] - D_{KL}\left(q(\mathbf{y}_l|\mathbf{X})||p(\mathbf{y}_l)\right)$$

Inference with VAEs

At the inference phase we can use one of the strategies:

- Sample. With this we'll get highly varying but uncontrollable intonation
- Fix mean. We will get intonation from a fixed cluster. By fixing y_o and thoroughly choosing the subset for mean we can control the style
- Zeros. Giving prior value sets AM to some "average" style. The output voice will be neutral, cleaned from all y_o-based features
- Predict from text. Predictor can be trained at a "good" subset of samples

GST

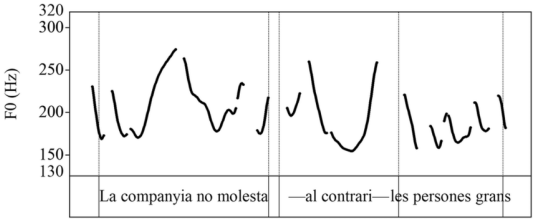
- Deterministic
- Uninterpretable space
- Can be made from text or ground truth

VAE

- Unstable training process
- Interpretable space
- Can incorporate known features into the space
- Can be made from text, ground truth or by averaging several GT examples

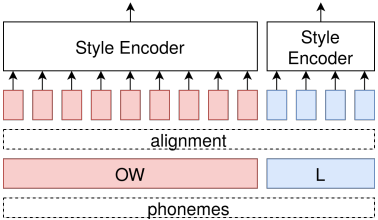
When Global Style isn't Enough

- Global style embeddings (both GST and VAE) cannot model particular words features
- For longer sentences number of possible style increases exponentially
- Intonation in many languages has a "concatenation-of-basic-blocks" structure
- So, we might want to control style locally, at phrase, word or syllable level



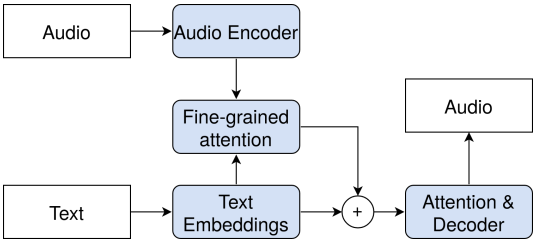
Utterance Segment Style Encoding

- We can use same style encoder technique at audio segment
- Timings can be extracted from ASR force-alignment or from AM teacher forcing
- We need very precise alignment especially if segments are small
- For segments we need more aggressive and lossy decoders - data leak with multiple style encoders is greater



[1] Link Fine-grained robust prosody transfer for single-speaker neural text-to-speech

Fine-grained Style Embedding



- Other approach - make model choose the proper segments itself
- We can add auxiliary attention to generate phoneme-wise styles:

$$\beta = \text{Attention}(E, A)$$

$$S = \beta^T E$$

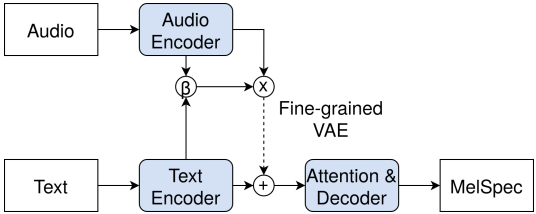
- A - audio frames embeddings, E - phonemes embeddings
- This embedding can be concatenated or added to text encoder output

[1] Link Robust and fine-grained prosody control of end-to-end speech synthesis

Fine-grained VAEs

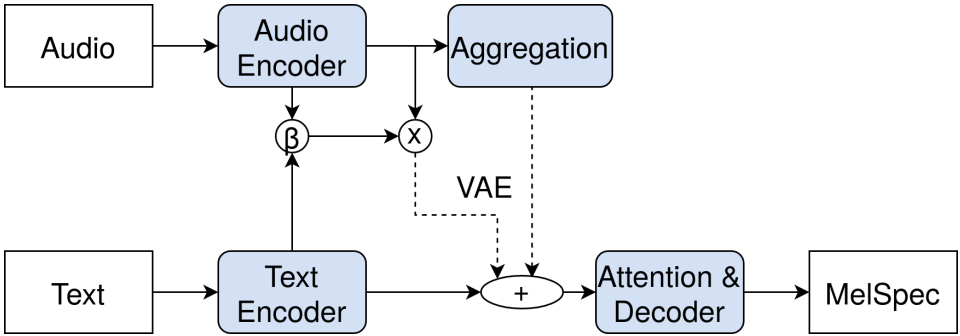
- We can apply the same VAE framework to the values S from fine-grained attention
- $\mathbf{z}_{ph}$ and $\mathbf{y}_{ph}$ can be utilized in the same loss formula

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z}|\mathbf{X})}\left[\log p(\mathbf{X}|\mathbf{Y}, \mathbf{z})\right] - \sum_i D_{KL}\left(q(\mathbf{z}_i^{ph}|\mathbf{X})||p(\mathbf{z}_i^{ph})\right)$$



Double Style Control

We can combine both approaches:



[1] Link Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis

Speech Styles

- We can control prosodic component of the speech information by adding style encoder
- We can use GST or VAE for style control
- GST is more deterministic and robust, VAE provides more interpretable style space
- We can use force-alignment timings or fine-grained attention mechanism to control style individually on phonemes
- In contemporary models both style control mechanisms are applied: global and local

Break!

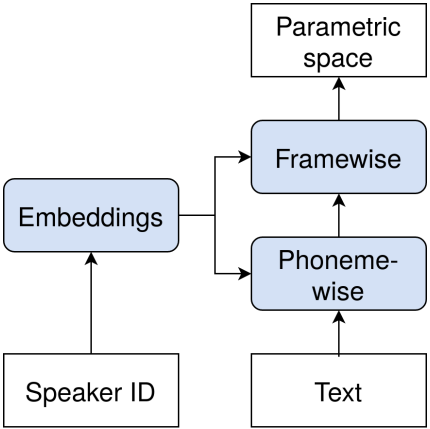
10 min

Many Speakers

Limits of Mono-speaker Corpus

- Hard to collect (Tacotron2 requires 15-20 hours of data to create a stable and natural voice)
- Hard to add new features (when the contract with voice talent is over)
- No few- and zero-shot
- A new model for each new speaker

Multi-speaker AM



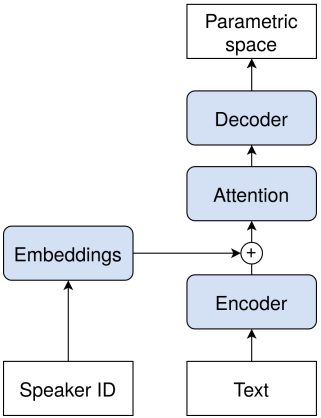
- Let's just add speaker embedding to all the nets
- Reduces amount of data needed
- Applicable to LSTM+Upsample, Attention-based and FFT+Upsample models
- Excessive conditioning — ineffective sharing of the model parts

[1] Link Deep Voice 2: Multi-Speaker Neural Text-to-Speech

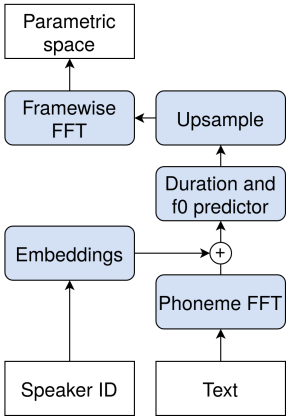
Speaker Information

- AM parts that depend on speaker info:
  - Duration modeling
  - Speaking style predictor
  - Frame-level signal modeling
- So, encoder of tacotron can be shared between speakers
- As well as the first FFT in fast-pitch/fast-speech

Speaker Information



Multi-speaker Tacotron



Multi-speaker Fast-Pitch

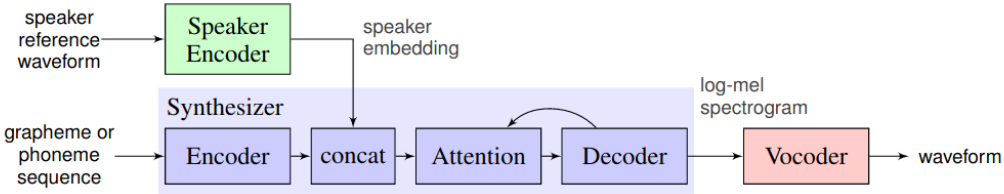
Speaker Information

With this multi-speaker setup:

- We need less data from a speaker (up to several minutes for nice and stable voice)
- One model can handle multiple speakers at once
- Articulation data and other linguistical features are shared across the speaker set
- We has a close set of speakers and need to re-train model after adding a new one

Zero-shot Speaker Leaning

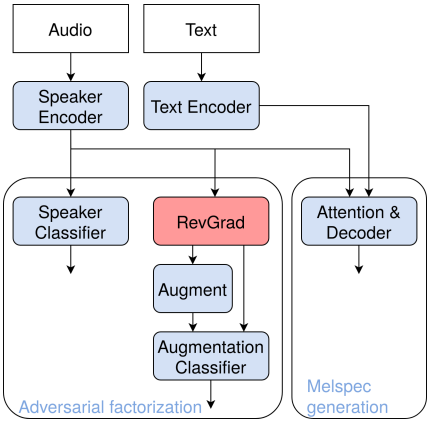
- We can use speaker verification encoding[1] instead of embedding
- With the train speaker set wide enough it allows us to copy new voice from a few samples
- Speaker encoder here — a pretrained d-vector[2] neural net



[1] Link Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis

[2] Link Deep neural networks for small footprint text-dependent speaker verification

Disentangle Speaker Info



- Speaker info can correlate with other, unwanted information (noise, bad recording quality etc.)
- To disentangle such relation we can apply reverse gradient
  1. Add randomly augmentation, similar to unwanted feature (e.g. noise)
  2. Add classifier: if speaker embedding is augmented or not
  3. Reverse the classifier gradient
- The loss will be:

$$\mathcal{L} = \mathcal{L}_{mel} + \mathcal{L}_{spk} - \mathcal{L}_{augment}$$

[1] Link Disentangling Correlated Speaker and Noise for Speech Synthesis via Data Augmentation and Adversarial Factorization

Multi-speaker

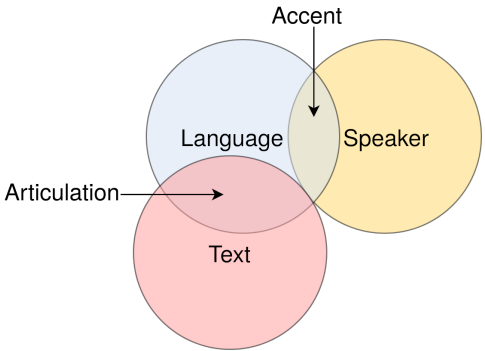
- We can add speaker embedding to LSTM+Upsample, Attention-based and FFT+Upsample AM architectures
- Speaker info is only needed in duration modeling and acoustic features prediction
- By adding speaker info we can reduce amount of data per single speaker
- By replacing speaker embedding with biometry verification model embedding we can make AM clone voice from a few samples
- With an adversarial loss we can split speaker info from other correlated features (useful for cleaning noisy data)

More Languages

Multi-language AM

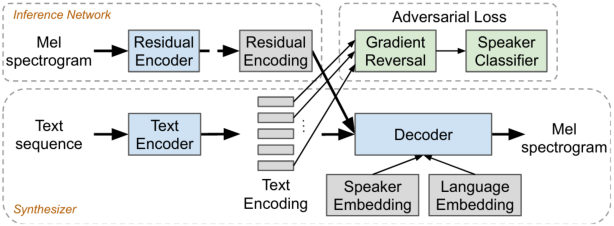
- Main idea: share model between languages like between speakers
- We want speaker to speak at unfamiliar language
- We want to make speaker talk with or without foreign accent

Speaker, Language and Text



- Speaker features are strongly connected with language information
- It is hard to disentangle textual (articulation of certain phonemes) and prosodic (accent) information in language
- We need special subnets to do these disentanglements

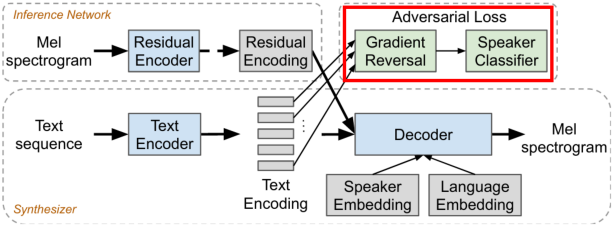
Multi-language Tacotron



The architecture:

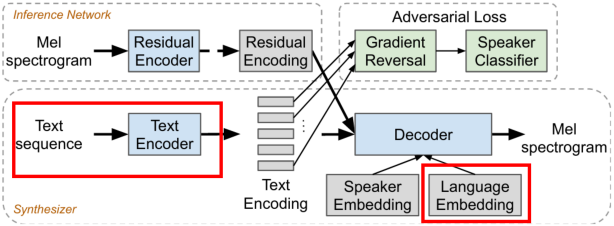
- Can model speaker in an unfamiliar language
- Has VAE-based style control
- Can separately control text- and prosodic-level language information

[1] Link Learning to Speak Fluently in a Foreign Language



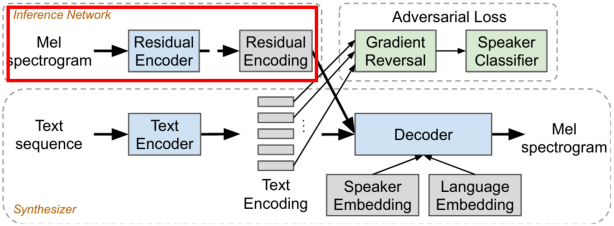
- The model has encoder separate from speaker features
- Separation provided by the adversarial speaker classifier loss
- If trainset has multiple speakers per language this separation prevents model from speaker morphing

[1] Link Learning to Speak Fluently in a Foreign Language



- Two language inputs:
  - Language embedding inside the decoder controls the prosodic component of the language information
  - Language-specific text tokens control the textual component
- By sending different information to these inputs we can model accents

[1] Link Learning to Speak Fluently in a Foreign Language



- Style control provided by the residual encoder
- That is VAE-based control of the style
- Feeding the prior mean (zeros) in the inference phase provides better inter-language speaker transfer

[1] Link Learning to Speak Fluently in a Foreign Language

Multi-language

- Without any additional measures multi-language model will suffer from speaker morphing when transferring voice between languages
- To prevents this we need adversarial speaker loss on text-level parts of the model
- By giving model different information in text tokens and language embedding we can model accents

End of Lecture