COM4511 Speech Technology: Speech Synthesis

Anton Ragni

March 11, 2020

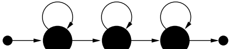
SPandH

Why Speech Synthesis?

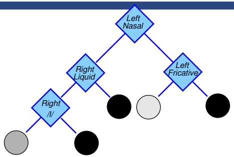
- One of key interfaces of speech technology
  - dialogue systems, assistants, synthetic voices and many more!
- Can be regarded as an inverse problem to speech recognition
  - generate speech given observed latent representation (word sequence)
- Previous lectures examined multiple generative models
  - why do we need to talk about speech synthesis?

Anton Ragni

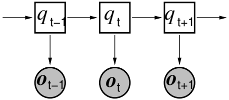
Hidden Markov Models



(a) Standard ASR topology



(b) Phonetic decision trees



(c) Dynamic Bayesian Network

- Probability density function of observation sequences given word sequence

$$p(\boldsymbol{O}_{1:T}|\boldsymbol{w}_{1:L}) = \sum_{\boldsymbol{q}_{1:T} \in \boldsymbol{Q}_{1:T}^{(\boldsymbol{w}_{1:L})}} p(\boldsymbol{O}_{1:T}|\boldsymbol{q}_{1:T})P(\boldsymbol{q}_{1:T}|\boldsymbol{w}_{1:L})$$

- Simple form of conditional density if state output distributions are Gaussians

$$p(\boldsymbol{O}_{1:T}|\boldsymbol{q}_{1:T}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}) = \mathcal{N}(\boldsymbol{O}_{1:T}; \boldsymbol{\mu}_{\boldsymbol{q}_{1:T}}, \boldsymbol{\Sigma}_{\boldsymbol{q}_{1:T}})$$

  - "synthesise speech" by maximising likelihood or sampling (how?)
  - what is the form of $\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}_{1:T}}$?
- BUT ...

Anton Ragni

Dynamic Features

$$\begin{bmatrix} \vdots \\ \boldsymbol{o}_{t+1} \\ \boldsymbol{o}_t \\ \boldsymbol{o}_{t-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \dots & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \dots & -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \dots \\ \dots & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \dots \\ \dots & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \dots \\ \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \dots \\ \dots & \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{o}_{t+1}^{(s)} \\ \boldsymbol{o}_t^{(s)} \\ \boldsymbol{o}_{t-1}^{(s)} \\ \vdots \end{bmatrix}$$

- Introduce dynamic features to relax conditional independence assumption

$$\boldsymbol{o}_t = \begin{bmatrix} \boldsymbol{o}_t^{(s)} \\ \Delta \boldsymbol{o}_t^{(s)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{o}_{t+1}^{(s)} \\ \boldsymbol{o}_t^{(s)} \\ \boldsymbol{o}_{t-1}^{(s)} \end{bmatrix}$$

  - possible to incorporate higher order "derivatives" (used by all non-NN HMMs)
- Yield linear transformation of the underlying speech parameterisation (e.g. MFCC)

$$\boldsymbol{O}_{1:T} = \boldsymbol{W}\ \boldsymbol{O}_{1:T}^{(s)}$$

  - any statistical model defined over $\boldsymbol{O}_{1:T}$ rather than $\boldsymbol{O}_{1:T}^{(s)}$ is inconsistent (inc. HMM)
  - derive sequence mean and covariance for static distribution $\mathcal{N}(\boldsymbol{O}_{1:T}^{(s)}; \boldsymbol{\mu}_{\boldsymbol{q}_{1:T}}^{(s)}, \boldsymbol{\Sigma}_{\boldsymbol{q}_{1:T}}^{(s)})$

Anton Ragni

Distribution of Static Features

- Express distribution of static features

$$\frac{1}{Z_{\boldsymbol{q}_{1:T}}}\mathcal{N}(\boldsymbol{O}_{1:T};\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}},\boldsymbol{\Sigma}_{\boldsymbol{q}_{1:T}}) = \frac{1}{Z_{\boldsymbol{q}_{1:T}}}\mathcal{N}(\boldsymbol{W}\boldsymbol{O}^{(s)}_{1:T};\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}},\boldsymbol{\Sigma}_{\boldsymbol{q}_{1:T}})$$

$$\propto \quad \exp\left(-\frac{1}{2}(\boldsymbol{W}\boldsymbol{O}^{(s)}_{1:T}-\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}(\boldsymbol{W}\boldsymbol{O}^{(s)}_{1:T}-\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}})\right)$$

$$\propto \quad \exp\left(-\frac{1}{2}\left(\boldsymbol{O}^{(s)\mathsf{T}}_{1:T}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{W}\boldsymbol{O}^{(s)\mathsf{T}}_{1:T} - \boldsymbol{\mu}^{\mathsf{T}}_{\boldsymbol{q}_{1:T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{W}\boldsymbol{O}^{(s)}_{1:T} - \boldsymbol{O}^{(s)\mathsf{T}}_{1:T}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}} + \boldsymbol{\mu}^{\mathsf{T}}_{\boldsymbol{q}_{1:T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}}\right)\right)$$

$$= \quad \mathcal{N}\left(\boldsymbol{O}^{(s)}_{1:T};\left(\boldsymbol{W}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{\mu}_{\boldsymbol{q}_{1:T}},\left(\boldsymbol{W}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{q}_{1:T}}\boldsymbol{W}\right)^{-1}\right) = \mathcal{N}(\boldsymbol{O}^{(s)}_{1:T};\boldsymbol{\mu}^{(s)}_{\boldsymbol{q}_{1:T}},\boldsymbol{\Sigma}^{(s)}_{\boldsymbol{q}_{1:T}}) = p(\boldsymbol{O}^{(s)}_{1:T}|\boldsymbol{q}_{1:T})$$

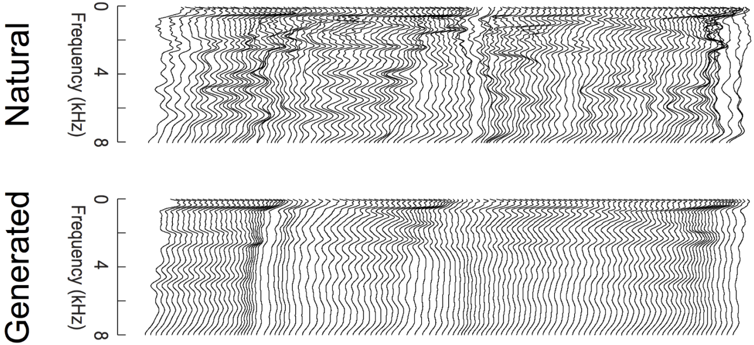
  - complex mean and covariance structure breaks conditional independence assumptions
  - BUT training preserves these assumptions (inconsistency)
- Maximise probability distribution of static features during training (Trajectory HMM)

$$p(\boldsymbol{O}^{(s)}_{1:T}|\boldsymbol{w}_{1:L}) = \sum_{\boldsymbol{q}_{1:T}\in\boldsymbol{Q}^{(\boldsymbol{w}_{1:L})}_{1:T}} p(\boldsymbol{O}^{(s)}_{1:T}|\boldsymbol{q}_{1:T})P(\boldsymbol{q}_{1:T}|\boldsymbol{w}_{1:L})$$

  - training and inference expensive due to dependency on the whole sequence

SPandH

Smoothness



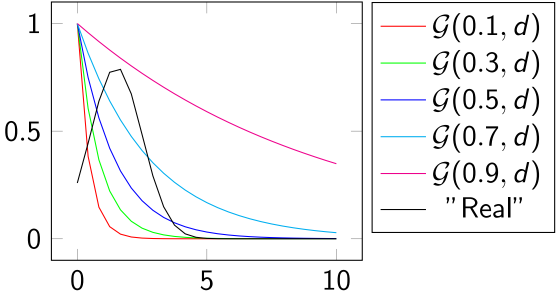
- Synthesised speech sounds "muffled" (low amplitude broadened formants)
- Enforce expected variance on generated speech

$$\hat{\boldsymbol{O}}^{(s)}_{1:T} = \arg\max_{\boldsymbol{O}^{(s)}_{1:T}} \left\{ \mathcal{N}\left(\boldsymbol{O}^{(s)}_{1:T}, \boldsymbol{\mu}^{(s)}_{\boldsymbol{q}_{1:T}}, \boldsymbol{\Sigma}^{(s)}_{\boldsymbol{q}_{1:T}}\right) \mathcal{N}\left(\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{o}^{(s)}_{t} - \boldsymbol{\mu})^2; \boldsymbol{\mu}^{(gv)}, \boldsymbol{\Sigma}^{(gv)}\right)^{\alpha} \right\}$$

  - global variance statistics, $\boldsymbol{\mu}^{(gv)}$ and $\boldsymbol{\Sigma}^{(gv)}$, estimated on training data
  - what does it remind you of?

Anton Ragni

Duration Modelling



- Probability of latent variable sequence in left-to-right topology

$$P(\boldsymbol{q}_{1:T}|\boldsymbol{w}_{1:L}) = a_{1,2}^{d_{1,2}} a_{2,2}^{d_{2,2}} a_{2,3}^{d_{2,3}} \dots a_{K-1,K-1}^{d_{K-1,K-1}} a_{K-1,K}^{d_{K-1,K}}$$

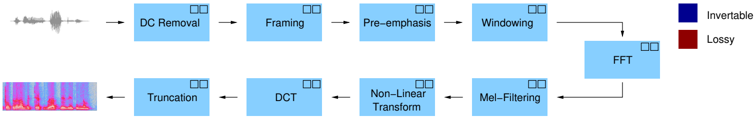
  - transition probability $a_{i,j}$ and count $d_{i,j}$
- Probability of staying d times in state j is proportionate to

$$P(q_t \neq j, q_{t+1} = j, q_{t+2} = j, \dots, q_{t+d} = j, q_{t+d+1} \neq j) \propto a_{j,j}^d = \mathcal{G}(a_{j,j}, d)$$

  - a geometric distribution — very poor model of phone durations
- Need a proper duration distribution — Hidden Semi-Markov Model

Anton Ragni

Speech Parameterisations

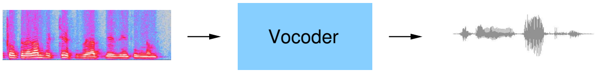


Mel Frequency Cepstral Coefficients (European Standards Organisation ETSI ES 201 108)

- Standard speech parameterisations are lossy
  - fundamental frequency F0 (pitch) filtered out
  - phase removed due to perceived indifference to human ear
- Augment spectral features with needed information
  - care needed due to non-trivial nature of pitch and phase
  - what have you learnt about pitch and phase?
- Example: colour code MFCC blocks as invertable/non-invertable, lossy/lossless

Anton Ragni

Vocoder



- Standard speech parameterisations are not invertable
  - impossible to recover the original waveform
- Multiple options available:
  - iterative Griffin-Lim algorithm

$$\boldsymbol{C}_{1:T}^{(n+1)} = \left(\boldsymbol{G}\boldsymbol{G}^{\dagger}\left(\boldsymbol{A}_{1:T} \odot \exp(\boldsymbol{j}\angle \boldsymbol{C}_{1:T}^{(n)})\right)\right), \quad \boldsymbol{s}_{1:fT} = \boldsymbol{G}^{\dagger}\boldsymbol{C}_{1:T}^{(N)}$$

    - complex spectrum $\boldsymbol{C}_{1:T}$ (amplitude $\boldsymbol{A}_{1:T}$, phase $\angle\boldsymbol{C}_{1:T}$), $\boldsymbol{G}$ STFT, $\boldsymbol{G}^{\dagger}$ inverse STFT
    - low-quality phase reconstruction algorithm
  - advanced signal processing techniques (MELP, STRAIGHT)
  - neural networks provide powerful alternative (later)

Modelling Units

Type	ASR	TTS	Description
Neighbours	✓	✓	previous/following phones
Position	✗[†]	✓	within syllable, word, phrase
Stress/Accent	✗[†]	✓	degree, "stress" distances
Linguistic Role	✗	✓	POS
Suprasegmental	✗	✓	length, tone

Contextual information usage in ASR and TTS

- Standard choices of modelling units in speech recognition
  - phones (/aa/, /ah/, ..., /zh/), graphemes (a, b, ..., y, z)
  - co-articulation effects handled by context-dependent units (name each type)

    `aa-v, v+dh, aa-v+dh, aa^aa-v+dh=ax`

  - BUT no expressive characteristic is explicitly modelled — poor synthesis quality
- Increase specificity of phonetic units

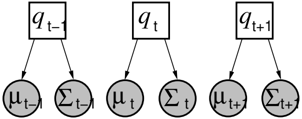
  ```
  aa^aa-v+dh=ax @ 2_1 / A:1_0_1 / B:1-0-2@1-1&11-3#9-2$2-1!1-2;8-2|aa / C:0+0+2 /
  D:content_1 / E:in+1@10+3&7+1#1+2 / F:det_1 / G:0_0 / H:13=12@1=1|L-L% / I:0=0 /
  J:13+12-1
  ```

  - decision trees enable robust estimates BUT do fragment available data (options?)

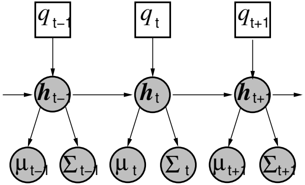
[†] often used with "limited" resource (non-English) languages

Anton Ragni

Mixture Density Networks



(a) Pseudo DBN DNN (TTS)



(b) Pseudo DBN RNN (TTS)

- Predict Gaussian mixture model given current (DNN) or all past (RNN) latent variables

$$p(\boldsymbol{o}_t | q_t, \ldots, q_1) = \sum_{m=1}^{M} c_{t,m} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{t,m}, \boldsymbol{\Sigma}_{t,m})$$

  - mean and variance parameters

$$\boldsymbol{\mu}_{t,m} = \phi^{(m)}(\boldsymbol{A}^{(m)}\boldsymbol{h}_t + \boldsymbol{b}^{(m)}), \quad \text{vec}(\boldsymbol{\Sigma}_{t,m}) = \phi^{(v)}(\boldsymbol{A}^{(v)}\boldsymbol{h}_t + \boldsymbol{b}^{(v)})$$

    - constraints required to ensure positive semi-definite covariance matrices

Text Normalisation

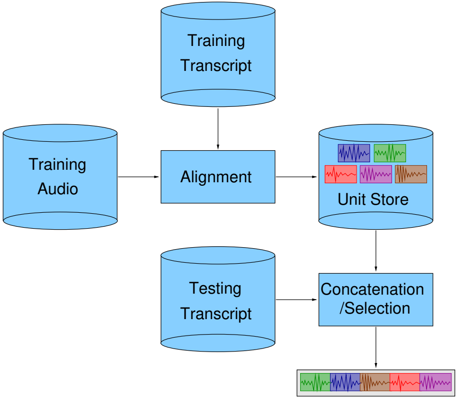
- Role varies among applications
  - ASR ("law" of large numbers)
    - one case, "small" vocabulary, optional number and abbr. expansion, punct. removal
  - TTS ("law" of small numbers)
    - preserve as much information as possible
- Examples:

St.	$\longrightarrow$	Street
St.	$\longrightarrow$	Saint
Ms	$\longrightarrow$	Miss
MS	$\longrightarrow$	Marks and Spencer
MS	$\longrightarrow$	Microsoft
£1984	$\longrightarrow$	one thousand, nine hundred and eighty four pounds
1984	$\longrightarrow$	nineteen eighty four
1984	$\longrightarrow$	one thousand nine hundred eighty four

- Accurate text normalisation is context-dependent

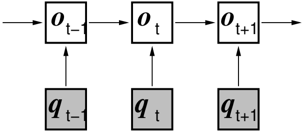
Anton Ragni

Concatenative Speech Synthesis



- Align transcripts to audio to create a unit store
  - need to decide on the nature of units, alignment process and audio representation
- (If necessary) select and concatenate units to "synthesise" a waveform
  - need to decide on the nature of selection and concatenation

Unit Selecton and Concatenation



Discrete elements:

- latent "observations" $\boldsymbol{o}_t$ (unit store)
- observed "latent" variables $\boldsymbol{q}_t$ (lexical description)

- Total cost of audio representation for given word sequence

$$\pi(\boldsymbol{O}_{1:T}, \boldsymbol{w}_{1:L}) = \sum_{\boldsymbol{Q}_{1:T} \in \mathcal{Q}_{1:T}^{(\boldsymbol{w}_{1:L})}} \pi(\boldsymbol{O}_{1:T}, \boldsymbol{Q}_{1:T}, \boldsymbol{w}_{1:L}) = \sum_{\boldsymbol{Q}_{1:T} \in \mathcal{Q}_{1:T}^{(\boldsymbol{w}_{1:L})}} \prod_{t=1}^{T} \pi(\boldsymbol{q}_t, \boldsymbol{o}_t)\pi(\boldsymbol{o}_{t-1}, \boldsymbol{o}_t)$$

  - target cost $\pi(\boldsymbol{q}_t, \boldsymbol{o}_t)$, concatenation cost $\pi(\boldsymbol{o}_{t-1}, \boldsymbol{o}_t)$
  - what does this form remind you of?
- Use dynamic programming to infer optimal observation sequence

$$\hat{\boldsymbol{O}}_{1:T} = \arg\max_{\boldsymbol{O}_{1:T}} \left\{ \pi(\boldsymbol{O}_{1:T}, \boldsymbol{Q}_{1:T}, \boldsymbol{w}_{1:L}) \right\}$$

  - use "smart" approaches to reduce search complexity

Anton Ragni

- Use advanced sequence models to map linguistic units into waveforms

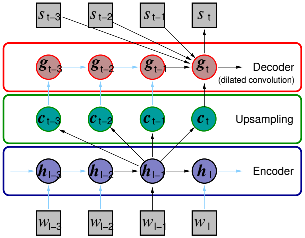
$$\boldsymbol{w}_{1:L} \longrightarrow \boldsymbol{s}_{1:fT}$$

  - (discrete) waveform samples $\boldsymbol{s}_{1:fT}$ typically taken at 16-24 kHz
  - what is f for 16 kHz and 24 kHz?
- Alternatively, introduce intermediate latent representation

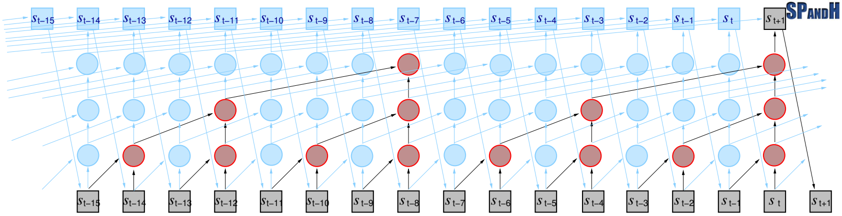
$$\begin{aligned} \boldsymbol{w}_{1:L} \longrightarrow \quad & \boldsymbol{Q}_{1:T} \\ & \boldsymbol{Q}_{1:T} \quad \longrightarrow \boldsymbol{s}_{1:fT} \end{aligned}$$

  - spectrogram $\boldsymbol{Q}_{1:T}$ provides interpretable representation
- In both cases need to decide if intermediate sub-word representation needed
  - "flat" or "rich" phonemes/graphemes

WaveNet



(a) Pseudo DBN



(b) Dilated Convolution

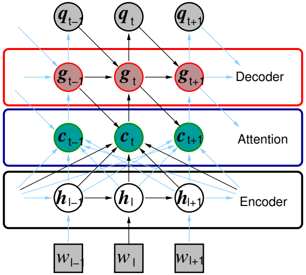
- Predict waveform conditioning on rich linguistic units

$$P(\boldsymbol{s}_{1:fT}|\boldsymbol{w}_{1:L}) \approx \prod_{t=1}^{fT} P(s_t|\boldsymbol{s}_{t-n+1:t-1}, \boldsymbol{w}_{1:L}) \approx \prod_{t=1}^{fT} P(s_t|\boldsymbol{g}_t)$$

  - dilated convolution handles high speech rate (16 kHz — 24 kHz)
  - upsampling layer handles low symbol rate (2 Hz words, 14 Hz phones)
- BUT autoregressive-like nature and high rate makes inference/training slow
  - additionally relies on carefully chosen linguistic units and fundamental frequency

Anton Ragni

Tacotron



Write down:

$$\begin{aligned} \boldsymbol{q}_t &= \dots \\ \boldsymbol{g}_t &= \dots \\ \boldsymbol{c}_t &= \dots \\ \boldsymbol{h}_l &= \dots \end{aligned}$$

- Predict spectrogram (100 Hz) conditioning on phone/grapheme sequence (14 Hz)

$$p(\boldsymbol{Q}_{1:T}|\boldsymbol{w}_{1:L}) = \prod_{t=1}^{T} p(\boldsymbol{q}_t|\boldsymbol{Q}_{1:t-1}, \boldsymbol{w}_{1:L}) \approx \prod_{t=1}^{T} p(\boldsymbol{q}_t|\boldsymbol{g}_t)$$

  - attention-based encoder-decoder architecture
- Predict waveform conditioning on spectrogram
  - use any suitable neural vocoder (e.g. WaveNet)

Anton Ragni

(Self-)Normalising Compositions

- Function composition (revisited)

$$\boldsymbol{y} = \boldsymbol{f}^{(K)} \odot \boldsymbol{f}^{(K-1)} \odot \ldots \odot \boldsymbol{f}^{(1)}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

  - if $p_{\mathcal{X}}(\boldsymbol{x})$ is know, which constraints $\boldsymbol{f}$ must obey to represent valid distribution $p_{\mathcal{Y}}(\boldsymbol{y})$
- Change of variables formula

$$p_{\mathcal{Y}}(\boldsymbol{y}) = p_{\mathcal{X}}(\boldsymbol{x}) \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}} \right) \right|$$

  - if $\boldsymbol{f}$ is invertible

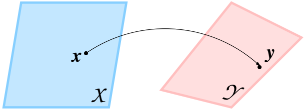
$$p_{\mathcal{Y}}(\boldsymbol{y}) = p_{\mathcal{X}}(\boldsymbol{x}) \left| \det \left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}^{-1} \right) \right| = p_{\mathcal{X}}(\boldsymbol{x}) \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|^{-1}$$

  - Jacobian of composition

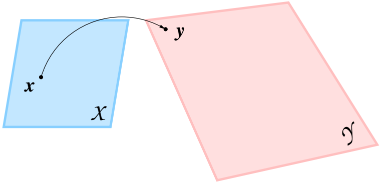
$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{f}^{(K)}}{\partial \boldsymbol{f}^{(K-1)}} \cdot \frac{\partial \boldsymbol{f}^{(K-1)}}{\partial \boldsymbol{f}^{(K-2)}} \cdot \ldots \cdot \frac{\partial \boldsymbol{f}^{(1)}}{\partial \boldsymbol{x}}$$

- Self-normalising compositions enable efficient sampling and training
  - provided Jacobian can be efficiently evaluated (conditions?)

Linear Transformation



(a) Rotation & Contraction



(b) Rotation & Expansion

- Linear transformation of Gaussian distributed random variable $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{X}}, \boldsymbol{\Sigma}_{\mathcal{X}})$

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$$

  - inverse transformation (assuming $\boldsymbol{A}$ invertable)

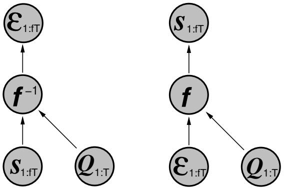
$$\boldsymbol{x} = \boldsymbol{f}^{-1}(\boldsymbol{y}) = \boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{b})$$

- Probability density of transformed variable as a function original density

$$p_{\mathcal{Y}}(\boldsymbol{y}) = p_{\mathcal{X}}(\boldsymbol{x}) \left| \det\left(\frac{\partial \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}{\partial \boldsymbol{x}} \right) \right|^{-1} = \frac{1}{|\boldsymbol{A}|} p_{\mathcal{X}}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{A}|} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathcal{X}}, \boldsymbol{\Sigma}_{\mathcal{X}})$$

  - compensate for contraction/expansion (discuss $\boldsymbol{A} = \mathrm{diag}(2)$, $\boldsymbol{A} = \mathrm{diag}(\frac{1}{2})$)

(Self-)Normalising Flows



(a) Training (b) Inference

Symbols:

$\epsilon_{1:fT}$ – *sample*
$\boldsymbol{Q}_{1:T}$ – *spectrogram*
$\boldsymbol{s}_{1:fT}$ – *waveform*
$\boldsymbol{f}$ – *invertable composition*

Invertable compositions:

- affine coupling
- 1×1 convolution
- many many more!

- Synthesise speech by passing noise (guess!?) through self-normalising compositions
  - training

$$p_{\mathcal{E}}(\boldsymbol{\epsilon}_{1:fT}) = p_{\mathcal{S}}(\boldsymbol{s}_{1:fT}) \left| \det \left(\frac{\partial \boldsymbol{f}^{-1}(\boldsymbol{s}_{1:fT}; \boldsymbol{Q}_{1:T})}{\partial \boldsymbol{s}_{1:fT}} \right) \right|^{-1}$$

  - inference

$$p_{\mathcal{S}}(\boldsymbol{s}_{1:fT}) = p_{\mathcal{E}}(\boldsymbol{\epsilon}_{1:fT}) \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{\epsilon}_{1:fT}; \boldsymbol{Q}_{1:T})}{\partial \boldsymbol{\epsilon}_{1:fT}} \right) \right|^{-1}$$

- Multiple normalising flows examined: inverse auto-regressive flow, WaveGlow

Anton Ragni

Affine Coupling

- Simple example of invertable composition

$$\boldsymbol{f}(\boldsymbol{\epsilon}_{1:fT}) = \begin{bmatrix} \boldsymbol{\epsilon}_{1:\tau} \\ \boldsymbol{\epsilon}_{\tau+1:fT} \odot \exp(\boldsymbol{f}^{(a)}(\boldsymbol{\epsilon}_{1:\tau})) + \boldsymbol{f}^{(b)}(\boldsymbol{\epsilon}_{1:\tau}) \end{bmatrix}, \quad \text{where} \quad \boldsymbol{\epsilon}_{1:fT} = \begin{bmatrix} \boldsymbol{\epsilon}_{1:\tau} \\ \boldsymbol{\epsilon}_{\tau+1:fT} \end{bmatrix}$$

  - scale $\boldsymbol{f}^{(a)}$ and bias $\boldsymbol{f}^{(b)}$ compositions need not be invertable
- Restricted form yields simple Jacobian

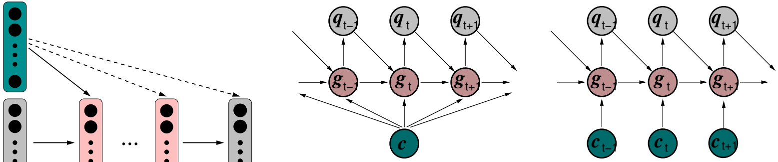
$$\frac{\partial \boldsymbol{f}(\boldsymbol{\epsilon}_{1:fT})}{\partial \boldsymbol{\epsilon}_{1:fT}} = \begin{bmatrix} \mathbf{1}_{1:\tau} & \mathbf{0} \\ \frac{\partial \boldsymbol{f}(\boldsymbol{\epsilon}_{1:fT})_{\tau+1:fT}}{\partial \boldsymbol{\epsilon}_{1:\tau}} & \text{diag}(\exp(\boldsymbol{f}^{(a)}(\boldsymbol{\epsilon}_{1:\tau}))) \end{bmatrix}$$

- Inverse composition

$$\boldsymbol{f}^{-1}(\boldsymbol{s}_{1:fT}) = \begin{bmatrix} \boldsymbol{s}_{1:\tau} \\ \left(\boldsymbol{s}_{\tau+1:fT} - \boldsymbol{f}^{(b)}(\boldsymbol{s}_{1:\tau})\right) \odot \exp(-\boldsymbol{f}^{(a)}(\boldsymbol{s}_{1:\tau})) \end{bmatrix}$$

- Example: prove that affine coupling is invertable $(\boldsymbol{f}^{-1}(\boldsymbol{f}(\boldsymbol{\epsilon}_{1:fT})) = \boldsymbol{\epsilon}_{1:fT}?)$

Conditioning



(a) Feed-Forward

1) Global

2) Temporal

(b) Recurrent

- Advanced models so far conditioned on simple symbol sequences (or nothing!)
  - may need to include other information (fundamental frequency, duration, speaker)
- Function composition makes conditioning simple
  - feed-forward unit

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{c})$$

  - recurrent unit

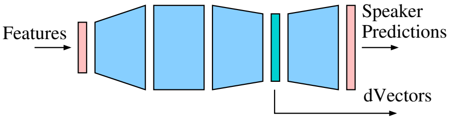
$$\boldsymbol{g}_t = \boldsymbol{f}(\boldsymbol{g}_{t-1}, \boldsymbol{q}_t; \boldsymbol{c}_t)$$

- BUT need to know how to create conditioning vectors

Speaker Representations

$$\phi^{(s)}(\boldsymbol{O}_{1:T}) = \begin{bmatrix} \log\left(\frac{p(\boldsymbol{O}_{1:T};\boldsymbol{\theta}^{(s)})}{p(\boldsymbol{O}_{1:T};\boldsymbol{\theta})}\right) \\ \nabla_{\boldsymbol{\theta}} \log(p(\boldsymbol{O}_{1:T};\boldsymbol{\theta}))|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(s)}} \end{bmatrix}$$

(a) Fisher kernel



(b) dVector

- Range of applications make use of speaker representations
  - speaker identification, verification, recognition, clustering, adaptation
- Typically continuous fixed dimensional representations

$$\boldsymbol{c}^{(s)} = \phi^{(s)}(\boldsymbol{O}_{1:T})$$

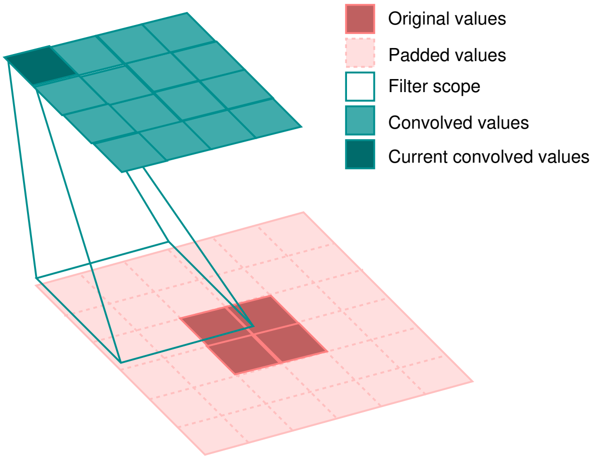
  - Fisher kernel, joint factor analysis, ?Vectors
- Consider neural network based dVector representation

$$\boldsymbol{c} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{h}_t^{(L-1)}$$

  - alternatively use non-uniform attention weights $\alpha_{1:T}$
  - what does dVector network topology reminds you of?

Anton Ragni

Up-Sampling



- Use transposed convolution ("de-convolution") to learn optimal up-sampling
  - options to choose filter, padding values, stride,

Training Criteria

- General training criterion — Minimum Mean Squared Error

$$\begin{aligned}\mathcal{L}(\mathcal{D};\boldsymbol{\theta}) &= -\frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\log(p(\hat{\boldsymbol{z}}_t^{(r)}|\hat{\boldsymbol{Z}}_{1:t-1}^{(r)},\hat{\boldsymbol{w}}_{1:L_r}^{(r)}))\\ &\propto -\frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}(\hat{\boldsymbol{z}}_t^{(r)}-\boldsymbol{z}_t^{(r)})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\hat{\boldsymbol{z}}_t^{(r)}-\boldsymbol{z}_t^{(r)})\end{aligned}$$

  - reference $\hat{\boldsymbol{z}}_t^{(r)}$ and model $\boldsymbol{z}_t^{(r)}$ prediction, noise covariance $\boldsymbol{\Sigma}$ (issues?)
- Alternatively, use classification style objective with discrete waveform samples

$$\mathcal{L}(\mathcal{D};\boldsymbol{\theta}) = -\frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T}\log\left(p(\hat{s}_t^{(r)}|\hat{\boldsymbol{s}}_{1:t-1}^{(r)},\hat{\boldsymbol{w}}_{1:L_r}^{(r)})\right)$$

  - discrete reference $\hat{s}_t^{(r)}$ and quantised model $\tilde{s}_t^{(r)}$ predictions (issues?)
  - quantise into 256 discrete values (why?) following μ-law compression

$$\tilde{s}_t^{(r)} = \text{int}\left(127\text{sign}(s_t^{(r)})\frac{\log(1+\mu|s_t^{(r)}|)}{\log(1+\mu)}+127\right)$$

Generative Adversarial Networks

- All generative models examined so far assume some parametric distribution
  - no matter how flexible not "true" distributions
  - in many cases unnecessary assumption!
- Often only samples needed so could forfeit ability to evaluate probability density
  - need to be able to judge samples as "good" or "bad" to train sample generator
- Pit sample generator against sample discriminator in a minimax "game"

$$\mathcal{L}(\boldsymbol{\theta}^{(d)}, \boldsymbol{\theta}^{(g)}) = \mathcal{E}_{\boldsymbol{O}\sim p_O}\left\{\log(d(\boldsymbol{O}; \boldsymbol{\theta}^{(d)}))\right\} + \mathcal{E}_{\boldsymbol{N}\sim p_N}\left\{\log(1 - d(\underbrace{g(\boldsymbol{N}; \boldsymbol{\theta}^{(g)})}_{\tilde{\boldsymbol{O}}}; \boldsymbol{\theta}^{(d)}))\right\}$$

  - true distribution p_O assumed to exist but not explicitly modelled (training data)
  - noise $\boldsymbol{N}$ sampled from noise distribution p_N converted to speech $\tilde{\boldsymbol{O}}$ by generator g
  - discriminator d classifies true and noise samples, generator makes this task harder
- BUT training such adversarial networks is not trivial
  - initialisation, stability, etc.

SPandH

Adversarial Synthesis

- One possible form of GAN objective function for sequence data

$$\mathcal{L}_{\texttt{gan}}(\mathcal{D};\boldsymbol{\theta}^{(d)},\boldsymbol{\theta}^{(g)}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\log(\sigma(d(\boldsymbol{o}_t^{(r)};\boldsymbol{\theta}^{(d)})))+\frac{1}{R'}\sum_{r'=1}^{R'}\sum_{t'=1}^{T'_{r'}}\log(1-\sigma(d(\tilde{\boldsymbol{o}}_{t'}^{(r')};\boldsymbol{\theta}^{(d)})))$$

  - samples can be drawn from parametric and nonparametric generators
  - average per-frame classification accuracy
- Incorporate GAN objective function in a soft fashion

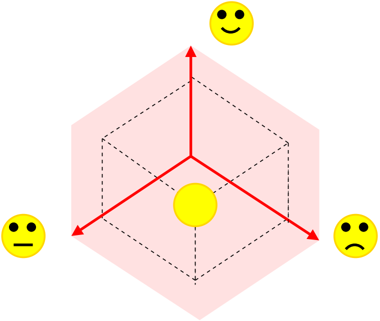
$$\mathcal{L}(\mathcal{D};\boldsymbol{\theta}^{(d)},\boldsymbol{\theta}^{(g)}) = \mathcal{L}_{\texttt{mge}}(\mathcal{D};\boldsymbol{\theta}^{(g)}) + \alpha\mathcal{L}_{\texttt{gan}}(\mathcal{D};\boldsymbol{\theta}^{(d)},\boldsymbol{\theta}^{(g)})$$

  - minimum generation error objective function

$$\mathcal{L}_{\texttt{mge}}(\mathcal{D};\boldsymbol{\theta}^{(g)}) = \frac{1}{R}\sum_{r=1}^{R}\|\boldsymbol{O}_{1:T_r}^{(r)} - \tilde{\boldsymbol{O}}_{1:T_r}^{(r)}\|_2^2$$

- Many other options available

Expressivity and Control



- Paralinguistic information crucial for naturalistic speech generation
  - emotions, speaking style, attitude
- Controlled modification complicated
  - not clear how to define expressive "state"
  - neural network adaptation challenging

Evaluation

- Objective measures
  - unlike speech recognition lacks commonly accepted measure
  - distortion (and other signal processing) style measures only indicative
- Subjective measures
  - Mean Opinion Score (MOS): Likert scale (0-5)
  - Preference Tests: better, worse, no preference
  - ABX Tests: distance to supplied reference (closer, further)
  - Transcription Test: transcribe what was said (and compare!)
- Issues with subjective measures
  - need to formulate precise criteria
  - need to ensure the criteria have been followed
  - can you name any other issue?

Summary

- These two lectures explored speech synthesis
  - hidden Markov models (HMM)
  - concatenative (unit selection) speech synthesis
  - advanced neural network approaches
  - evaluation
- Focus on issues surrounding generating realistic speech using HMMs
  - inconsistency
  - duration modelling
  - smoothness
  - vocoder
  - modelling units and text normalisation
- Also discussed advanced forms of neural networks
  - WaveNet and Tacotron
  - Normalising Flows
  - Generative Adversarial Networks

Anton Ragni