Voice Conversion using Deep Learning

Albert Aparicio Isarn

Advisors: Antonio Bonafonte and Santiago Pascual

Degree's Thesis Presentation, 23 May 2017

Outline

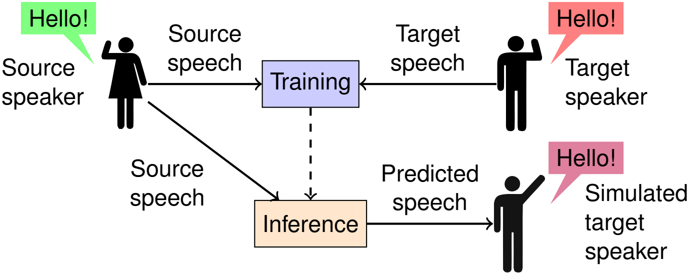
1. Background
   - Voice Conversion
   - Deep Learning
   - Data preparation
2. Proposed Models
   - Baseline
   - Sequence-to-Sequence Learning
   - Sequence-to-Sequence Models
3. Results/Contribution
   - Main Results
   - Future Work

Outline

1. Background
   - Voice Conversion
   - Deep Learning
   - Data preparation
2. Proposed Models
   - Baseline
   - Sequence-to-Sequence Learning
   - Sequence-to-Sequence Models
3. Results/Contribution
   - Main Results
   - Future Work

Main Objective

- Map features of source speaker to target speaker



Action figures credit: `freedesignfile.com/62581-action-figures-icons-vector-2`, CC-BY 3.0

Main Challenges

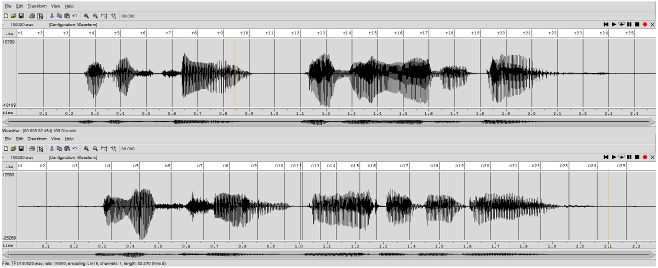
- Naturality
  - Make voice sound like a human
- Similarity
  - Make voice sound like the target speaker

Main Challenges

- Naturality
  - Make voice sound like a human
- Similarity
  - Make voice sound like the target speaker

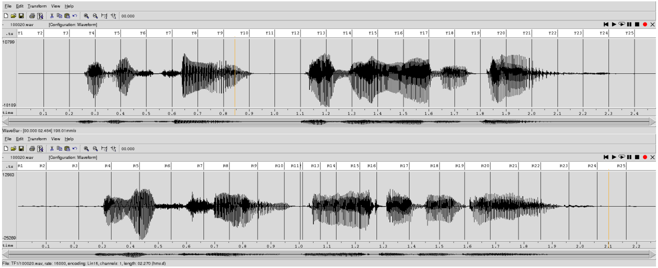
Common Techniques

- Classic Techniques
  - Gaussian Mixture Model (GMM)
  - Frequency Warping
- Deep Learning Techniques
- These techniques require aligned data
  - **Solution:** Sequence-to-Sequence learning



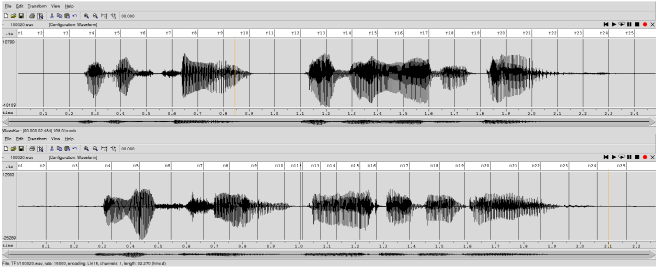
Common Techniques

- Classic Techniques
  - Gaussian Mixture Model (GMM)
  - Frequency Warping
- Deep Learning Techniques
- These techniques require aligned data
  - **Solution:** Sequence-to-Sequence learning



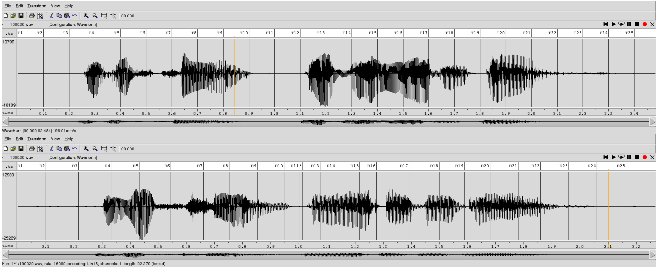
Common Techniques

- Classic Techniques
  - Gaussian Mixture Model (GMM)
  - Frequency Warping
- Deep Learning Techniques
- These techniques require aligned data
  - **Solution:** Sequence-to-Sequence learning



Common Techniques

- Classic Techniques
  - Gaussian Mixture Model (GMM)
  - Frequency Warping
- Deep Learning Techniques
- These techniques require aligned data
  - **Solution:** Sequence-to-Sequence learning



Outline

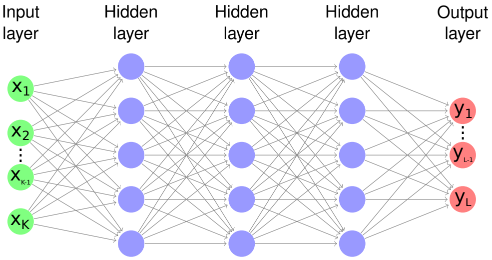
Deep Learning

"*A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification*" [1]

Main Strength - Ability to model complex non-linear mapping functions

[1] Li Deng and Dong Yu. Deep Learning: Methods and Applications. Tech Report MSR-TR-2014-21, NOW Publishers, Boston - Delft, May 2014. Pages 199-200

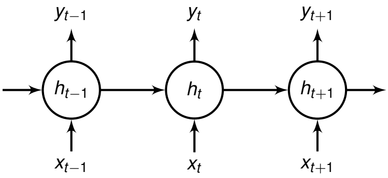
Feed-Forward Neural Network



- **Input** - $\boldsymbol{x} \in \mathbb{R}^K$
- **Output** - $\boldsymbol{y} \in \mathbb{R}^L$
- $\boldsymbol{W}$ - Weight matrix
- $\boldsymbol{b}$ - Bias vector
- $\boldsymbol{y} = f(\boldsymbol{W} \times \boldsymbol{x} + \boldsymbol{b})$

- $f \rightarrow$ Activation function - Allows DNN to model non-linearities
- Weights and biases trained with Back-propagation

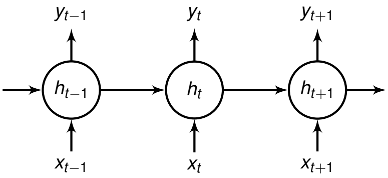
Recurrent Neural Network



- $h_t = f\left(W_{xh}\, x_t + W_{hh}\, h_{t-1} + b_h\right)$
- $y_t = f\left(W_{hy}\, h_t + b_y\right)$

- RNNs model temporal evolutions
- **Problems** - Vanishing and exploding gradients (training) and volatile memory (prediction)
  - **Solution** - LSTM, GRU or PLSTM cells

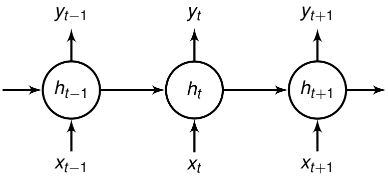
Recurrent Neural Network



- $h_t = f\left(W_{xh}\, x_t + W_{hh}\, h_{t-1} + b_h\right)$
- $y_t = f\left(W_{hy}\, h_t + b_y\right)$

- RNNs model temporal evolutions
- **Problems** - Vanishing and exploding gradients (training) and volatile memory (prediction)
  - **Solution** - LSTM, GRU or PLSTM cells

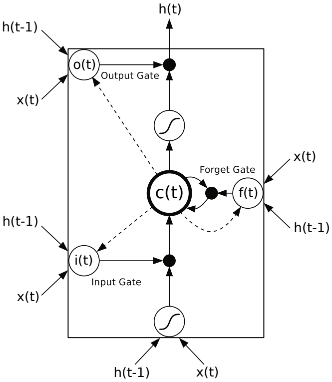
Recurrent Neural Network



- $h_t = f\left(W_{xh}\, x_t + W_{hh}\, h_{t-1} + b_h\right)$
- $y_t = f\left(W_{hy}\, h_t + b_y\right)$

- RNNs model temporal evolutions
- **Problems** - Vanishing and exploding gradients (training) and volatile memory (prediction)
    - **Solution** - LSTM, GRU or PLSTM cells

Long Short-Term Memory



- $\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right)$
- $\hat{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right)$
- $\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right)$
- $\boldsymbol{C}_t = \boldsymbol{i}_t \odot \hat{\boldsymbol{C}}_t + \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1}$
- $\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right)$
- $\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{C}_t\right)$

Figure credit: Graves, A., Supervised sequence labelling. Springer Berlin Heidelberg, 2012

From RNN to LSTM

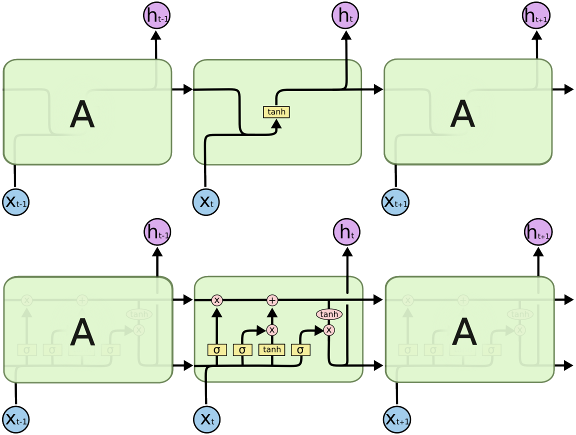


Figure credit: Olah, C., Understanding LSTM Networks, Accessed 22-05-2017,
`colah.github.io/posts/2015-08-Understanding-LSTMs`

Outline

Data preparation

- Datasets
  - Voice Conversion Challenge 2016
    - 10 speakers
    - 9 min/speaker
  - TC-STAR Dataset
    - 2 of the total speakers
    - 1.5h/speaker
- Data encoded with vocoder (Ahocoder)
  - Parameters
    - 40 Mel Cepstrum (MCP)
    - 1 log-Pitch (log f_0)
    - 1 Maximum Voiced Frequency (MVF)

Outline

Baseline Model

- Based off of the Interspeech 2016 proposal by Chen et al.
- Multiple Neural Networks
  - **GRU-RNN** - Mel Cepstrum parameters (MCP)
  - **LSTM-RNN** - log-Pitch (lf0)
  - **DNN** - Maximum Voiced Frequency (MVF)
- Training data from Voice Conversion Challenge (VCC) 2016

Alignment Problem

- Data usually aligned with Dynamic Time Warping (DTW) algorithm
- Needed by training to align frames with same speech
- **Problem** - Frame replication changes data statistics
    - **Solution** - Sequence-to-Sequence architecture

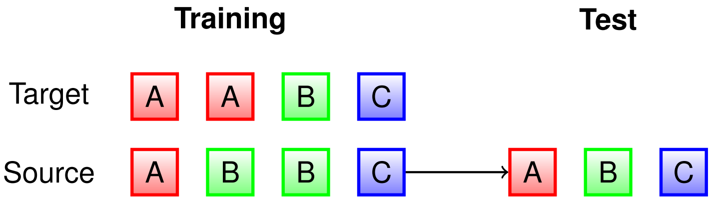
Alignment Problem

- Data usually aligned with Dynamic Time Warping (DTW) algorithm
- Needed by training to align frames with same speech
- **Problem** - Frame replication changes data statistics
    - **Solution** - Sequence-to-Sequence architecture

Alignment Problem

- Data usually aligned with Dynamic Time Warping (DTW) algorithm
- Needed by training to align frames with same speech
- **Problem** - Frame replication changes data statistics
  - **Solution** - Sequence-to-Sequence architecture

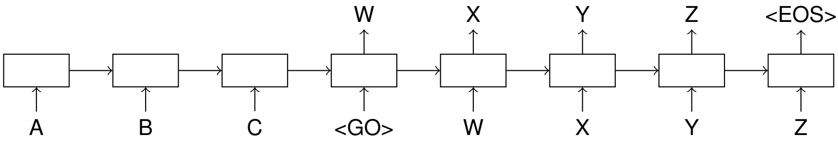
Alignment - Toy Example



Outline

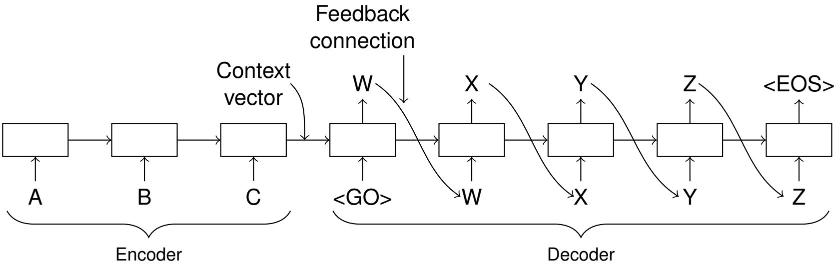
Sequence-to-Sequence Learning

- Proposed by Sutskever et al. in 2014
- Can work with sequences of different lengths
- Alignment is intrinsic to the model

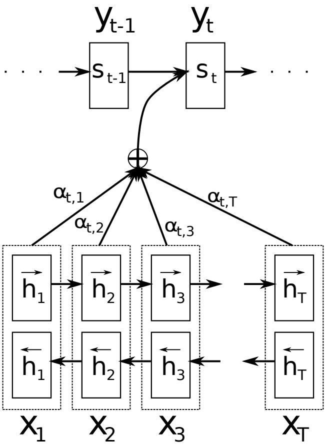


Sequence to Sequence Learning

- Encoder-Decoder architecture
- Feedback connection in each decoder timestep

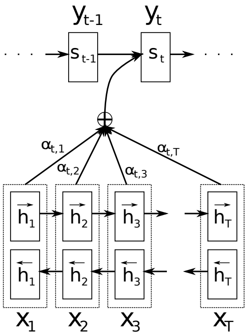


Attention Mechanism



- Proposed by Bahdanau et al. in 2014
- Improvement over Seq2Seq
- Each sequence timestep uses a different context vector

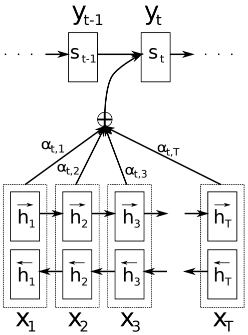
Attention Mechanism



- **Context vector is no longer constant**
- $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$
- $h_j \rightarrow$ annotation from the encoder at timestep j
- $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$
- $e_{ij} = a\left(s_{i-1}, h_j\right) \rightarrow$ alignment model

- Allows the decoder to select the relevant elements from the input sequence

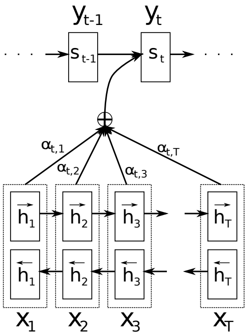
Attention Mechanism



- **Context vector is no longer constant**
- $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$
- $h_j \rightarrow$ annotation from the encoder at timestep j
- $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$
- $e_{ij} = a\left(s_{i-1}, h_j\right) \rightarrow$ alignment model

- Allows the decoder to select the relevant elements from the input sequence

Attention Mechanism



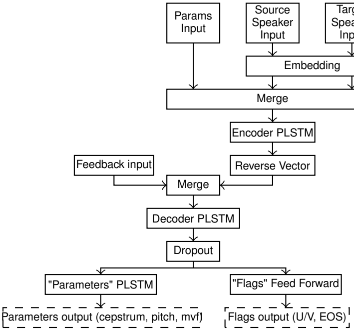
- **Context vector is no longer constant**
- $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$
- $h_j \rightarrow$ annotation from the encoder at timestep j
- $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$
- $e_{ij} = a\left(s_{i-1}, h_j\right) \rightarrow$ alignment model

- Allows the decoder to select the relevant elements from the input sequence

Outline

First implementations

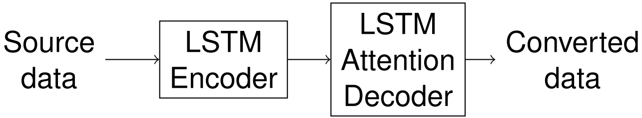
We have implemented multiple Seq2Seq implementations



- Keras - “Vanilla” Seq2Seq
- Keras - Seq2Seq with Feedback and Multiple Inputs and Outputs
  - Pretraining as autoencoder

Attention Seq2Seq Model

- Implementation from both TensorFlow and PyTorch
- LSTM Encoder - LSTM Decoder with Attention Mechanism
- Trained with speakers 75 and 76 from TC-STAR dataset



Outline

Baseline Results

Intelligible speech, although unnatural:

Source audio	Target audio	Converted audio
SF1 200005	TF1 200005	SF1 $\rightarrow$ TF1 200005
SF1 200012	TF1 200012	SF1 $\rightarrow$ TF1 200012
Stop		

Second Sequence-to-Sequence Results

Highly distorted signal. Does not sound like speech:

Source audio	Target audio	Converted audio
SF1 200005	TF1 200005	SF1 $\rightarrow$ TF1 200005
SF1 200012	TF1 200012	SF1 $\rightarrow$ SF1 200012
		SF1 $\rightarrow$ TF1 200012

Stop

Second Sequence-to-Sequence Pretraining

Ground truth data fed into the feedback loop gives Intelligible speech. Proves the problem is either the encoder or the feedback loop

Source audio

72 (SF1) 110167

72 (SF1) 200104

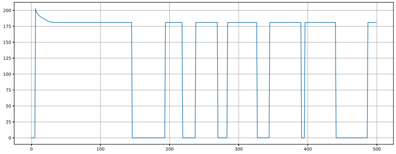
Stop

Autoencoded audio

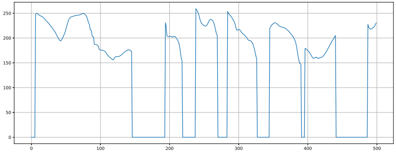
72 (SF1) 110167

72 (SF1) 200104

Attention Sequence-to-Sequence



(a) Predicted data



(b) Target data

Low variability of the predicted signal

Attention Sequence-to-Sequence

Ground truth cepstrum with predicted pitch and MVF

Source audio

75 (SF2) 330159

Stop

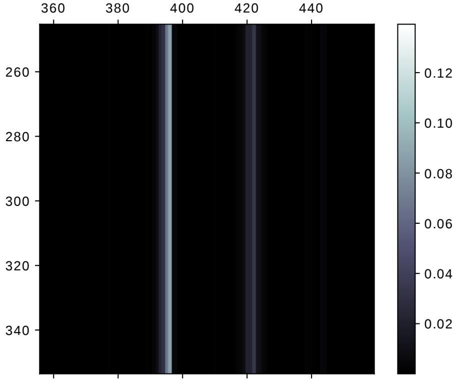
Target audio

76 (SF3) 330159

Converted audio

75 (SF2) $\rightarrow$ 76 (SF3) 330159

Attention Sequence-to-Sequence



Attention graph from PyTorch model

Bad alignment with the Attention Mechanism

"*One problem is that attention tends to get stuck for many frames before moving forward*" [Tacotron authors, 2017]

Outline

Future Work

- Investigate hypothesis of the poor results
  - Encoder is uncapable of mapping inputs to annotations
  - Attention Mechanism is not powerful enough to align data
- Efficient method for encoding long sequences

Summary

- First approach to solving unaligned voice conversion
- Contribution of new code to the Deep Learning community
  - `github.com/albertaparicio/tfg-voice-conversion`
  - `github.com/albertaparicio/tfglib`

For Further Reading I

- Chen, L., Liu, L., Ling, Z., Jiang, Y., Dai, L
  The USTC System for Voice Conversion Challenge 2016: Neural Network Based Approaches for Spectrum, Aperiodicity and F_0 Conversion
  Proc. Interspeech 2016, 1642–1646, 2016.
- Sutskever, I., Vinyals, O., Le, Q.V.
  Sequence to Sequence Learning with Neural Networks
  `arXiv:1409.3215`, 2014
- Bahdanau, D., Cho, K., Bengio, Y.
  Neural Machine Translation by Jointly Learning to Align and Translate
  `arXiv:1409.0473`, 2014

For Further Reading II

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., Saurous, R.A..
Tacotron: Towards End-to-End Speech Synthesis
arXiv:1703.10135, 2017