

CS 419(M)
Introduction to Machine Learning

Course Project

Voice Conversion

Team Members:

Nihal Singh- 150040015

Srivatsan Sridhar- 150070005

Arpan Banerjee- 150070011





Introduction





Problem Description

- Convert one person's (source) voice into another person's (target) voice
- **Goals :**
 - Similarity to the target voice
 - Naturalness of the converted speech
- Machine Learning to capture more features than only signal processing methods
- **Applications** - Automatic dubbing, translation



State Of The Art

- Waveform to phoneme (TIMIT) - Hierarchical maxout CNN + dropout, 16.5% PER
- Text to speech Synthesis - Deepmind's Wavenet, MOS 4.55
- End to end models have started using GANs
 - Voice Conversion Challenge 2018: Best MOS just above 4



Methodology

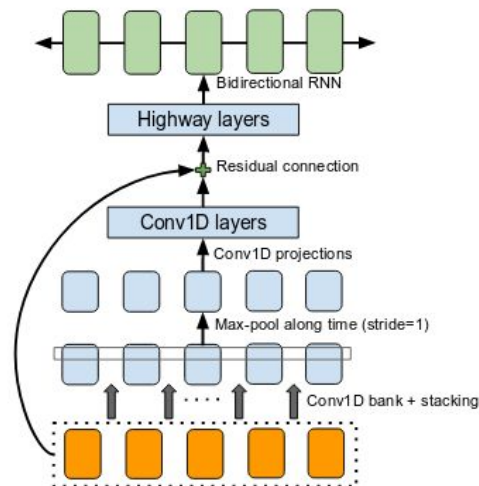




Literature Review and Inspiration

Tacotron - End to End Speech Synthesis (TTS)

- Introduced **CBHGs** (1-D Convolution Banks, Highway network and GRU) and **attention**
- Generates speech at the frame level, hence is substantially faster
- Uses a post-processing net and Griffin-Lim algorithm to generate speech audio
- Achieves MOS of 3.82 on US English with great results in terms of naturalness



Source: Tacotron: Towards End-to-End Speech Synthesis

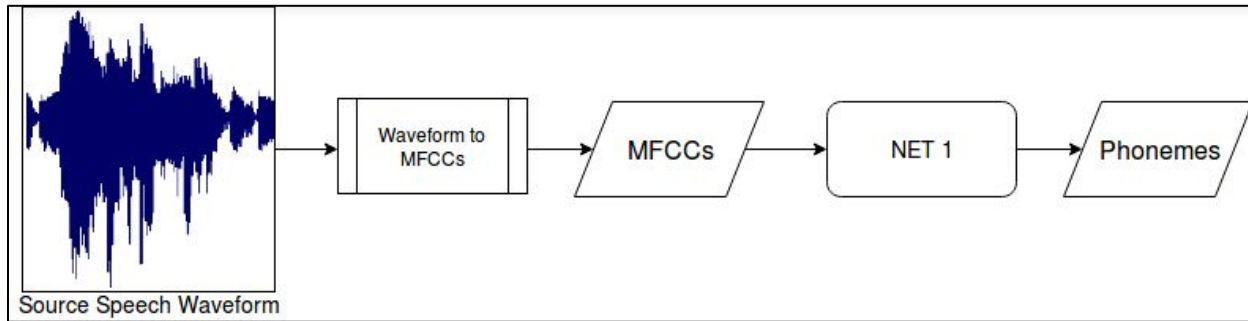


Dataset and Model Used

- **Net1** to convert source speech to phonemes
- Trained using **TIMIT** dataset - utterances of phonetically rich sentences by 630 speakers with frame level phoneme transcription

- **Net2** to convert phonemes to target speech
- Trained using **CMU Arctic** dataset - 593 train and 539 test samples of each individual speaker of varying accents and genders

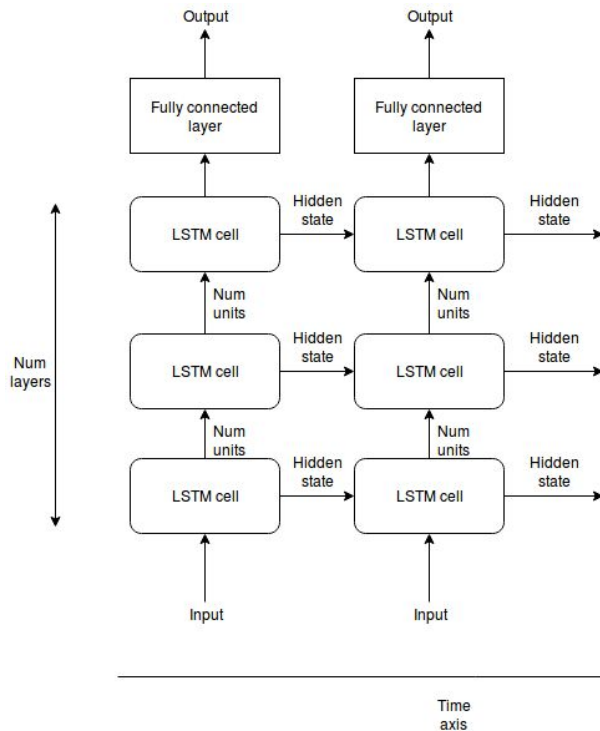
Net 1 - Source Speech to Phonemes



- Net 1 converts MFCCs to one phoneme per frame
- Long Short Term Memory (LSTM) cells stacked in an RNN
- Experimented with Unidirectional as well as Bidirectional configurations
- Further tested using Gated Recurrent Units (GRU) instead of LSTM
- Trained using Adam Optimizer for a Cross-Entropy Softmax Loss function

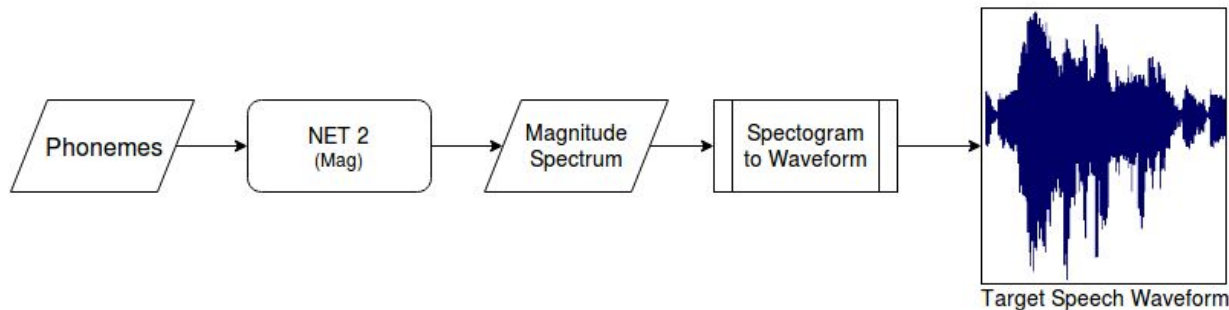


Net 1 - Source Speech to Phonemes



Explored the effects of changing number of layers and number of units in the RNN

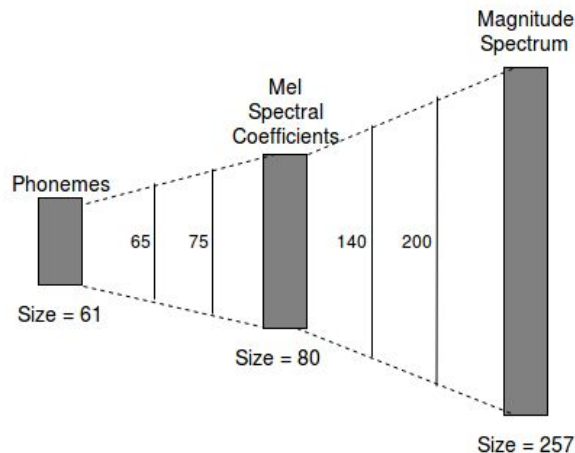
Net 2 - Phonemes to Target Speech



- Net 2 converts per-frame phonemes to log of STFT magnitudes
- Gated Recurrent Units (GRU) stacked in a bidirectional RNN
- Trained using Adam Optimizer on a Mean Squared Error loss
- Output magnitude spectrogram is converted to audio file using Griffin-Lim algorithm



Net 2 - Phonemes to Target Speech



- Pyramidal architecture
 - Number of units in each hidden layer increases from input to output
 - Results in lower Mean Squared Error
- Multitask Training
 - Train by evaluating loss over the intermediate mel-spectral coefficients along with the final magnitude spectrum



Experiments





Experiments

Experiments to test the effects of the following hyperparameters:

Parameters	Best
• Number of hidden layer	2
• Number of units per layer	200
• Dropout rate	0.8, 0.6
• GRU vs LSTM	close, occasionally outperform
• Pyramidal Structure	better
• Multitask Learning	similar results

Net 1

Unidirectional LSTM

Hidden Layers	Units per layer	Max. Test Accuracy (%)
1	50	68.3
	75	68.8
2	50	69.4
	75	70.0
	100	71.1
3	50	69.8
	75	70.2
	100	70.6
4	50	69.0
	75	71.1

Bidirectional LSTM

Hidden Layers	Units per layer	Keep probability	Max. Test Accuracy (%)
4	100	0.9	71.8
		0.8	72.5
		0.7	72.8
		0.6	72.3

Bidirectional GRU

Hidden Layers	Units per layer	Max. Test Accuracy (%)
1	50	67.9
	75	69.0
	100	69.4
2	50	69.9
	75	71.1
	100	70.3
3	50	70.4
	75	70.6
	100	70.8
4	50	70.7
	75	70.5

Bidirectional GRU and Bidirectional LSTM with dropout

Cell	Hidden Layers	Units per layer	Keep probability	Accuracy (%)
GRU	2	200	0.6	67.5
	3			62.2
	4			9.3
LSTM	2	200	0.6	73.5
	3			70.5
	4			69.4

Net 2



Net2

Bidirectional GRU

Hidden Layers	Units per layer	Keep probability	Mean Square Error
2	100	0.8	0.189
3			0.187
4			0.192
5			0.185

Bi-GRU with Pyramidal Structure

Units per layer	Keep probability	Mean Square Error
100, 150, 200	0.6	0.181
100, 150, 200	0.6	0.187
61, 124, 257	0.6	0.186

Bi-LSTM with Pyramidal Structure

Units per layer	Keep probability	Mean Square Error
100, 150, 200	0.6	0.164

Bi-GRU with Multitask (mags and mels)

Hidden Layers	Units per layer	Keep probability	Mean Square Error
2	100	0.6	0.383
3			0.391
4			0.388
5			0.392

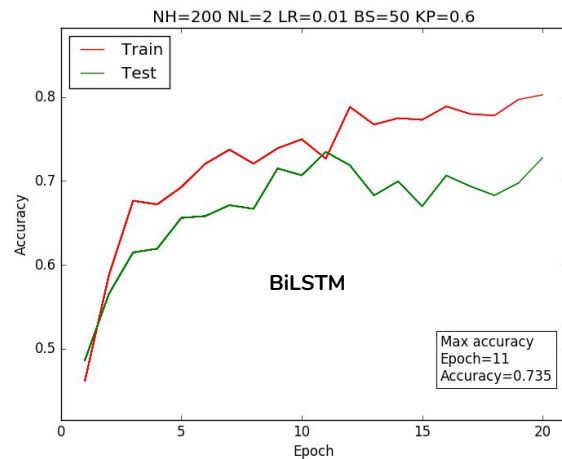
Bi-GRU with Pyramidal Structure and Multitask (mags and mels)

Num. units for mags	Num. units for mels	Keep probability	Mean Square Error
65, 75	140, 200	0.8	0.359
65, 75	140, 200	0.6	0.351
100, 100	100, 100	0.9	0.384

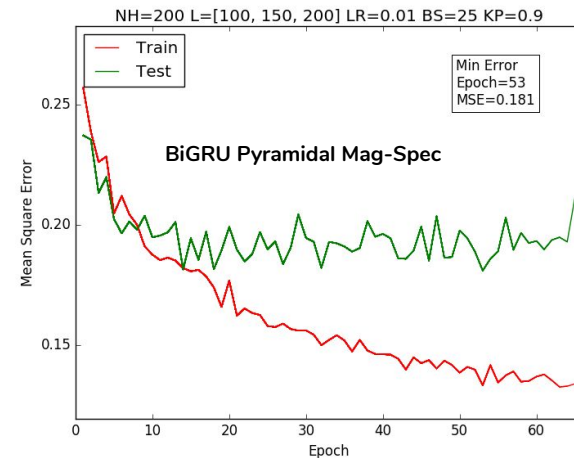
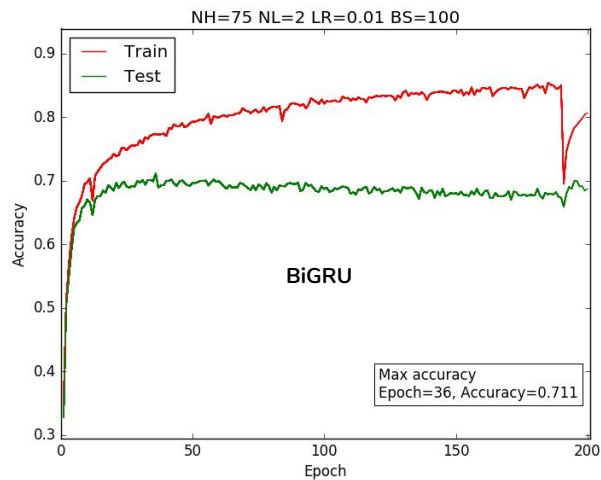
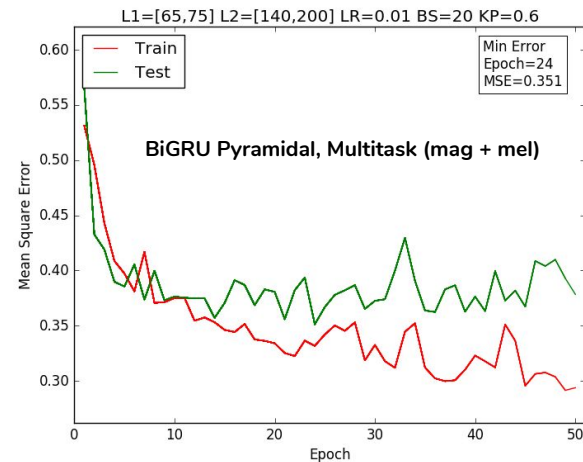
Bi-LSTM with Pyramidal Structure and Multitask (mags and mels)

Num. units for mags	Num. units for mels	Keep probability	Mean Square Error
65, 75	140, 200	0.6	0.546

Net 1



Net 2





Results

- Net 1 - Predicting phonemes
 - Best accuracy was 73.5%
 - This accuracy is a measure of per-frame phoneme classification instead of PER
 - Bidirectional LSTM (best = 73.5%) works slightly better than bidirectional GRU (best = 71.1%) for Net1
- Does well on clean American accent. More particularly, it performs well on the TIMIT and arctic test set.
- Quite a few of the misclassifications are due to predicting similar sounding (but different phonemes) in intermediate frames (for eg- 'ae' instead of 'aa').





Results

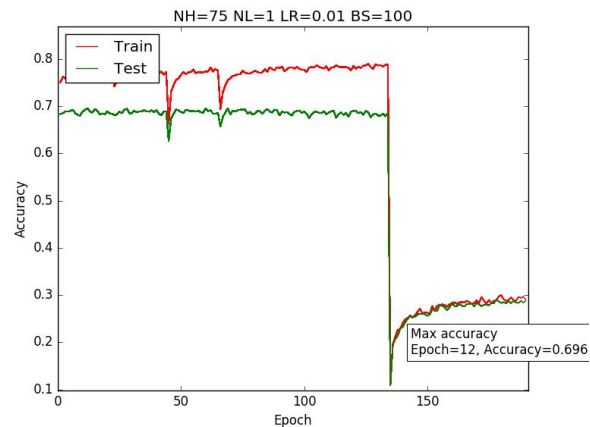
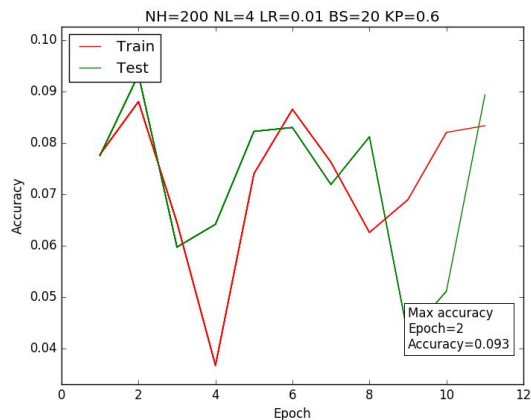
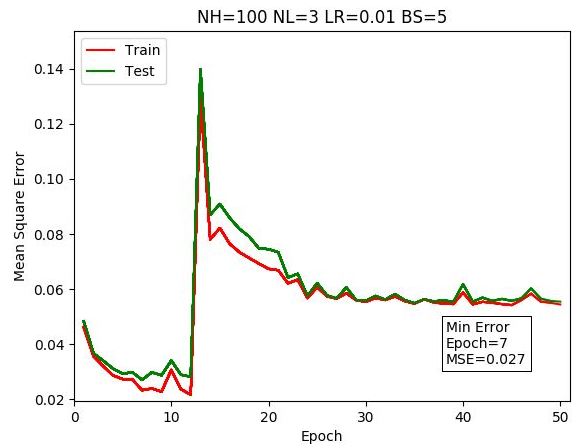
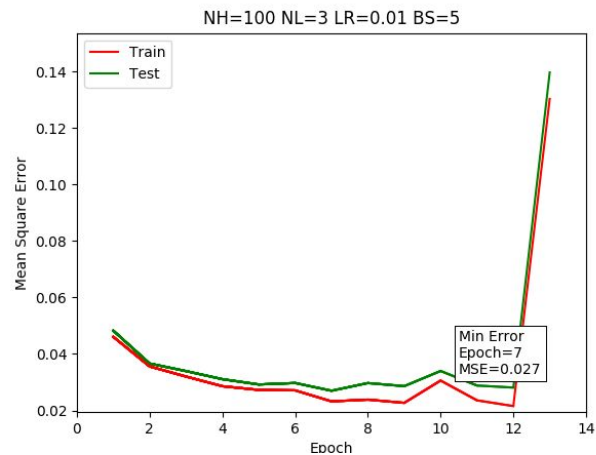
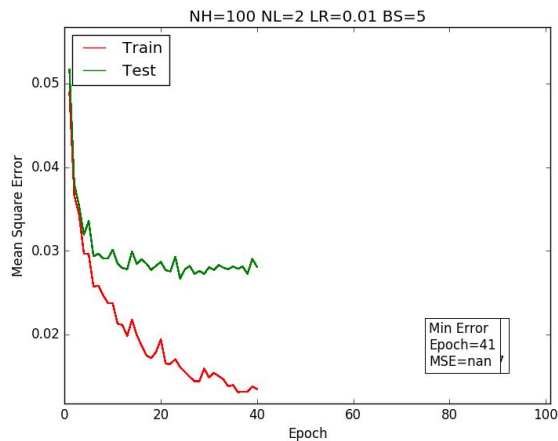
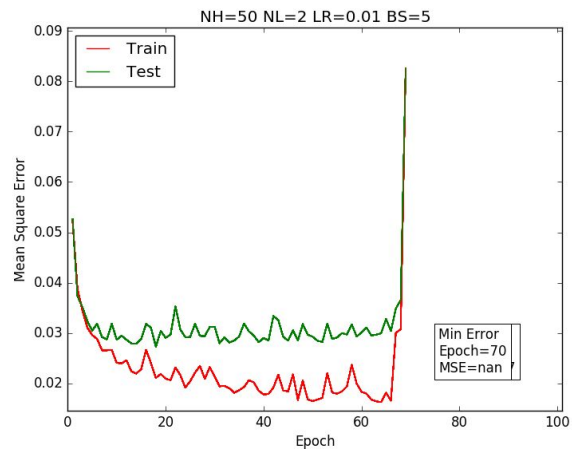
- Net 2 - Generating target speech
 - Lowest MSE on normalized target magnitude spectrum was 0.18
 - Lowest MSE on the pyramidal structure with multitask (sum of normalized mel spectral and mag. spectral errors) was 0.35
- We found that the accuracy of Net1 didn't matter as significantly as the accuracy of Net2 did while converting a voice sample end-to-end.



End-to-End Results

- Reconstruction of audio from magnitude spectrum
- We find that the reconstructed audio sounds very close to the original
- Original  Reconstructed
- End-to-end voice conversion
- Source  Target
- The resulting audio sounds robotic because of limitations in phoneme to spectrogram conversion.

The fault in our graphs



Conclusion



Success of our voice conversion model :

- Succeeds in retaining the intelligibility of the spoken sentence
- Retains some features of the target speaker

Limitations of our voice conversion model :

- Unnatural quality of generated voice
- Phoneme to spectrogram conversion performs poorly

Future Work

- Explore deeper models like Tacotron to give better results
- Training Net2 to directly predict the complex STFT - magnitude and phase

Thank You

