Large-Scale Mixed-Bandwidth Deep Neural Network Acoustic Modeling for ASR

Khoi-Nguyen C. Mac*, Xiaodong Cui, Wei Zhang and Michael Picheny

*Department of ECE, UIUC, USA
IBM Research AI, IBM T. J. Watson Research Center, USA

09/16/2019

Outline

- Experimental investigation of mixed-band (MB) acoustic modeling
- Strategies for MB acoustic modeling
  - Downsampling
  - Upsampling
  - Bandwidth extension (BWE)
    - Investigate a discriminatively trained BWE scheme
- Experimental results
  - Large-scale training data with unbalanced amounts of WB (1,150h) and NB (2,300h) speech
  - Diverse test sets from a variety of real-world application domains
- Summary and future work

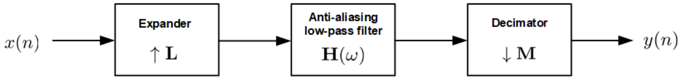
Why Mixed-Band Acoustic Modeling Is Appealing

- Both WB and NB speech signals widely exist in speech applications
  - WB: broadcast news
  - NB: telephony speech
- WB and NB acoustic models are usually separately trained for ASR
- One acoustic model for both WB and NB would be great for real-world system deployment

How To Carry Out Mixed-Band Acoustic Modeling

- The goal of MB acoustic modeling is to converge WB and NB speech to one bandwidth
- Common strategies
  - Downsampling
  - Upsampling
  - BWE
- Interested in seeking answers to the following questions:
  1. Which strategy is better, upsampling or downsampling?
  2. How would direct pooling perform under DNNs?
  3. How would BWE help in this case?

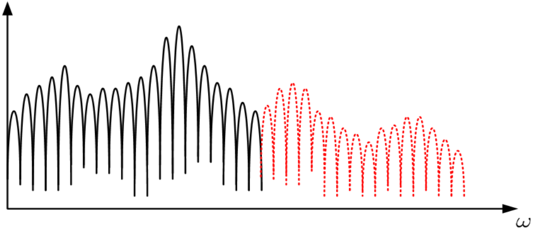
Downsampling and Upsampling



Sampling rate $$\Omega_y = \frac{L}{M}\Omega_x$$

- Classical multirate signal processing
- Typically carried out in the time domain

Bandwidth Extension for ASR (1)



- Estimates missing high frequency spectral components
- Has been extensively studied in communication and acoustic processing for a long time.
- Usually aims to improve intelligibility and quality of perception

Bandwidth Extension for ASR (2)

Problem Formulation:

- $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ denotes a sequence of n NB features
- $\hat{\mathbf{Y}} = \{\hat{\boldsymbol{y}}_1, \cdots, \hat{\boldsymbol{y}}_n\}$ denotes a sequence of n WB features
- Establish a mapping f_θ with parameter θ: $\hat{\mathbf{Y}} = f_\theta(\mathbf{X})$

Common approaches:

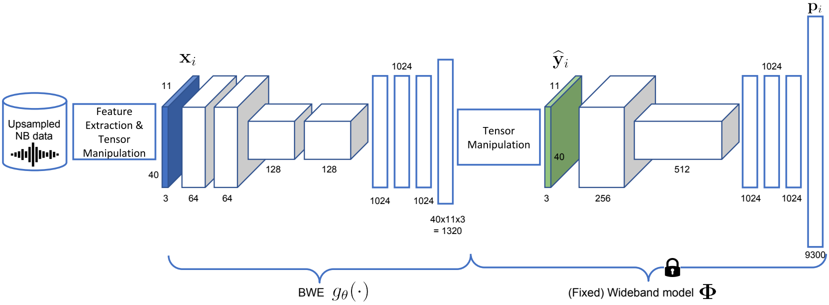
- Treated as a regression problem (parallel data required)

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \| \boldsymbol{y}_i - f_\theta(\boldsymbol{x}_i) \|_2^2$$

- Treated as a generative problem
  - e.g. generative adversarial nets (GANs)

Caveat: They may not be well aligned with the ASR performance

A Discriminatively Trained BWE



- Discriminatively trained BWE $\quad \theta^* = \text{argmin}_\theta \frac{1}{n} \sum_{i,k} l_{ik} \log \frac{1}{p_{ik}}$
- Fixed WB acoustic model
- Labels generated by aligning upsampled NB speech against WB acoustic model
- Optimization of BWE more related to ASR performance

Training Data

- 1,150 hours WB speech
  - 420h broadcast news data
  - 450h internal dictation data
  - 100h meeting data
  - 140h hospitality (travel and hotel reservation) data
  - 40h accented data
- 2,300 hours NB speech
  - 2,000h Switchboard
  - 300h IBM call-center data

Test Data and Decoding

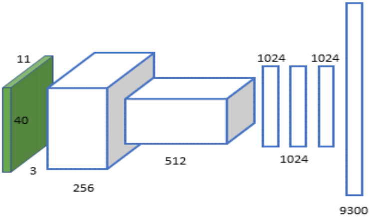
- 4 WB test sets and 4 NB test sets
- 4-gram LM consisting of 200M n-grams, trained on a broad variety of sources
- 250K decoding vocabulary

		Description	Hours
WB	WS1	Dev04f test set from Broadcast News	2.21
	WS2	Commercial services help desk	0.34
	WS3	Hospitality domain 1	1.21
	WS4	Hospitality domain 2	0.81
NB	NS1	Hub5-2000 test set from Switchboard	2.10
	NS2	Technical support	4.09
	NS3	Commercial services help desk	3.01
	NS4	Multi-domain command and control	12.78

System Implementation

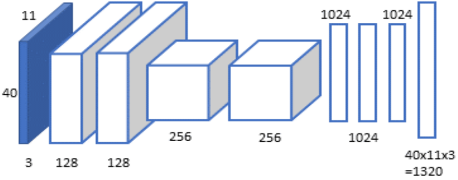
- Models
  - CNN acoustic models for WB, NB and MB
  - VGG-like CNN models for BWE
- Feature Space
  - 16KHz for WB, 8KHz for NB
  - Upsampling and downsampling carried out in time domain
  - 40-dim logmel, Δ, Δ^2, temporal context of 11 frames
  - Global CMN followed by utterance-based CMN
- Distributed Training
  - Synchronous data parallel training on 8 Nvidia v100 GPUs
  - Allreduce based on NCCL

CNN Acoustic Models



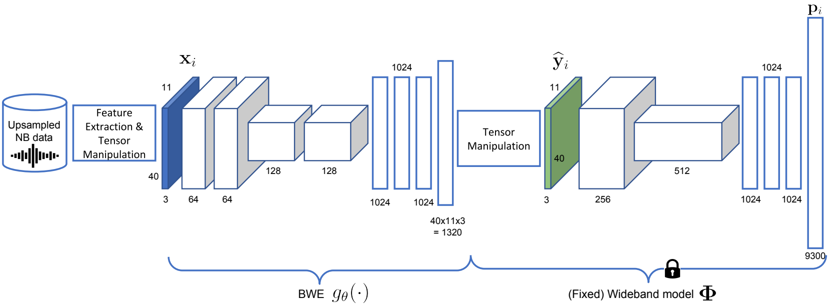
- CNN models for both WB and NB speech
- 2 conv layers, each followed by a max-pooling layer
- kernel 5×5, stride 1×1 and padding 2×2 for conv layers
- kernel 2×2 and stride 2×2 for max-pooling layers
- Relu activation except sigmoid for the last FC layer
- two capacities with (128,256) and (256,512) feature maps respectively

CNN BWE Models



- VGG-like architecture
- 4 conv layers and a max-pooling layer after every 2 conv layers
- kernel 3×3, stride 1×1 and padding 1×1 for conv layers
- kernel 2×2 and stride 2×2 for max-pooling layers
- Relu activation except tanh for the last FC layer
- two capacities with (64,128) and (128,256) feature maps respectively

A Discriminatively Trained BWE



Experiments – An Overview

(1) WB and NB baselines

(2) Direct pooling of WB and upsampled NB

(3) BWE decoded against WB model

(4) Pooling of WB and NB after BWE

(5) Fine-tuning by alternated optimization of BWE and MB models

Performance of Direct Pooling

	WB					NB				
	WS1	WS2	WS3	WS4	**Avg**	NS1	NS2	NS3	NS4	**Avg**
WB baseline ([128,256])	15.4	14.9	9.1	29.2	**17.2**	<u>25.1</u>	<u>39.0</u>	<u>13.7</u>	<u>22.0</u>	**<u>25.0</u>**
NB baseline ([128,256])	<u>21.3</u>	<u>16.8</u>	<u>15.6</u>	<u>40.5</u>	**<u>23.6</u>**	13.5	25.0	12.8	19.7	**17.8**
WB+NB↑,[128,256])	17.1	13.0	12.2	27.9	**17.6**	13.8	25.5	12.2	19.6	**17.8**
WB+NB↑,[256,512])	16.5	12.8	11.8	28.8	**17.5**	13.4	25.2	11.8	19.2	**17.4**
WB↓+NB,[128,256])	18.9	17.2	13.3	35.9	**21.3**	14.0	26.2	12.5	19.1	**18.0**

- Sampling rate mismatch gives rise to significant degradation
- Direct pooling helps
- Increasing model capacity helps
- Upsampling performs better than downsampling under pooling

Performance of BWE

	WB	NB
WB baseline ([128,256])	17.2	<u>25.0</u>
NB baseline ([128,256])	<u>23.6</u>	17.8
BWE ([64,128])	-	18.9
BWE ([128,256])	-	18.6
nBWE ([64,128])	-	18.7

- BWE can significantly improve upsampled NB speech against WB acoustic model
- Increasing model capacity of BWE helps
- Improvement is consistent across test sets
- Denoising BWE helps

Performance of Pooling with BWE

	WB					NB				
	WS1	WS2	WS3	WS4	**Avg**	NS1	NS2	NS3	NS4	**Avg**
WB+NB↑,[128,256]	17.1	13.0	12.2	27.9	**17.6**	13.8	25.5	12.2	19.6	**17.8**
WB+NB↑,[256,512]	16.5	12.8	11.8	28.8	**17.5**	13.4	25.2	11.8	19.2	**17.4**
WB↓+NB,[128,256]	18.9	17.2	13.3	35.9	**21.3**	14.0	26.2	12.5	19.1	**18.0**
WB+NB↑+BWE, [128,256]	16.5	14.2	10.1	29.9	**17.7**	13.6	25.6	12.2	19.7	**17.8**
WB+NB↑+BWE, [256,512]	16.0	14.6	9.7	29.9	**17.6**	13.7	25.4	12.2	19.6	**17.7**
WB+NB↑+nBWE, [128,256]	16.4	14.3	10.0	30.9	**17.9**	13.7	25.6	12.1	19.5	**17.7**

- BWE model sticks to model capacity of [64,128]
- Improves from BWE alone
- Slightly better than direct pooling under the same model capacity
- No improvements from direct pooling with large capacity

Performance of Fine-tuning of Pooling with BWE

	WB	WB
WB+NB↑+BWE, [128,256]	17.7	17.8
WB+NB↑+BWE, [256,512]	17.6	17.7
WB+NB↑+nBWE, [128,256]	17.9	17.7
WB+NB↑+BWE+FT, [128,256]	17.6	17.9
WB+NB↑+BWE+FT, [256,512]	17.6	17.8
WB+NB↑+nBWE+FT, [128,256]	17.7	17.7

– BWE CNN is connected to the (fixed) MB CNN
– Finetune with a smaller learning rate
– Training another MB CNN
– No consistent improvement.

Summary and Future Work

- It is possible to train a MB model of competitive performance
  - Upsampling appears to be more helpful than downsampling
- Direct pooling WB and upsampled NB with appropriately increased model capacity gives good performance
  - the MB model yields lower average WERs over NB baseline with only slight degradation over WB baseline
- BWE helps upsampled NB data against WB model
  - Pilot experiments show that discriminatively trained BWE outperforms MMSE-based BWE
- No strong observation that pooling WB and NB with BWE is better than direct pooling under increased model capacity
  - No consistent gains across a broad variety of test sets
  - Although direct pooling assumes no explicit BWE, DNNs with sufficient capacity may implicitly learn the mapping during training
- Looking forward
  - More powerful deep generative model with discriminative training

Complete Experimental Results

	WB					NB				
	WS1	WS2	WS3	WS4	**Avg**	NS1	NS2	NS3	NS4	**Avg**
WB baseline ([128,256])	15.4	14.9	9.1	29.2	**17.2**	<u>25.1</u>	<u>39.0</u>	<u>13.7</u>	<u>22.0</u>	**<u>25.0</u>**
NB baseline ([128,256])	<u>21.3</u>	<u>16.8</u>	<u>15.6</u>	<u>40.5</u>	**<u>23.6</u>**	13.5	25.0	12.8	19.7	**17.8**
DirectMix (WB+NB↑,[128,256])	17.1	13.0	12.2	27.9	**17.6**	13.8	25.5	12.2	19.6	**17.8**
DirectMix (WB+NB↑,[256,512])	16.5	12.8	11.8	28.8	**17.5**	13.4	25.2	11.8	19.2	**17.4**
DirectMix (WB↓+NB,[128,256])	18.9	17.2	13.3	35.9	**21.3**	14.0	26.2	12.5	19.1	**18.0**
BWE ([64,128])	-	-	-	-	-	15.2	27.8	12.4	20.2	**18.9**
BWE ([128,256])	-	-	-	-	-	14.9	27.4	12.2	20.0	**18.6**
nBWE ([64,128])	-	-	-	-	-	15.0	27.6	12.4	19.6	**18.7**
Mix (WB+NB↑+BWE, [128,256])	16.5	14.2	10.1	29.9	**17.7**	13.6	25.6	12.2	19.7	**17.8**
Mix (WB+NB↑+BWE, [256,512])	16.0	14.6	9.7	29.9	**17.6**	13.7	25.4	12.2	19.6	**17.7**
Mix (WB+NB↑+nBWE, [128,256])	16.4	14.3	10.0	30.9	**17.9**	13.7	25.6	12.1	19.5	**17.7**
MixFT (WB+NB↑+BWE, [128,256])	16.6	14.8	9.9	29.2	**17.6**	13.6	25.6	12.5	19.7	**17.9**
MixFT (WB+NB↑+BWE, [256,512])	16.1	15.1	9.7	29.3	**17.6**	13.7	25.5	12.4	19.6	**17.8**
MixFT (WB+NB↑+nBWE, [128,256])	16.2	14.4	9.8	30.3	**17.7**	13.6	25.4	12.0	19.6	**17.7**