Deep Learning for Speech Recognition

Hung-yi Lee
Outline

• Conventional Speech Recognition
• How to use Deep Learning in acoustic modeling?
• Why Deep Learning?
• Speaker Adaptation
• Multi-task Deep Learning
• New acoustic features
• Convolutional Neural Network (CNN)
• Applications in Acoustic Signal Processing
Conventional Speech Recognition
Machine Learning helps

Speech Recognition

This is a structured learning problem.

Evaluation function: \( F(X, W) = P(W | X) \)

Inference: \( \tilde{W} = \arg \max_W F(X, W) = \arg \max_W P(W | X) \)

\[
\begin{align*}
\tilde{W} &= \arg \max_W \frac{P(X|W)P(W)}{P(X)} \\
&= \arg \max_W P(X|W)P(W)
\end{align*}
\]

\( P(X|W) \): Acoustic Model, \( P(W) \): Language Model
Input Representation

- Audio is represented by a vector sequence

\[ x_1, x_2, x_3, \ldots, \quad \text{39 dim MFCC} \]
Input Representation - Splice

- To consider some temporal information
Phoneme

• Phoneme: basic unit

Each word corresponds to a sequence of phonemes.

Lexicon

what do you think

Lexicon

Different words can correspond to the same phonemes
State

• Each phoneme correspond to a sequence of states

what do you think

Phone:

hh w aa t d uw y uw th ih ng k

Tri-phone:

...... t-d+uw d-uw+y uw-y+uw y-uw+th ......

t-d+uw1 t-d+uw2 t-d+uw3 d-uw+y1 d-uw+y2 d-uw+y3

State:
State

• Each state has a stationary distribution for acoustic features

Gaussian Mixture Model (GMM)

\[ P(x|"t-d+uw1") \]

\[ P(x|"d-uw+y3") \]
State

- Each state has a stationary distribution for acoustic features.
Acoustic Model

\[ \hat{W} = \arg \max_W P(X|W)P(W) \]

\( W: \) what do you think?

\( S: \) a b c d e ......

\( X: \) x1 x2 x3 x4 x5 ......

Assume we also know the alignment \( s_1 \cdots s_T \).

\[ P(X|S, h) = \prod_{t=1}^{T} P(s_t|s_{t-1})P(x_t|s_t) \]

transition emission
Acoustic Model

\[ \hat{W} = \arg \max_W P(X|W)P(W) \]

**W:** what do you think?

**S:** a b c d e ......

Actually, we don’t know the alignment.

**X:** x₁ x₂ x₃ x₄ x₅ ......

\[ P(X|S) \approx \max_{s_1 \ldots s_T} \prod_{t=1}^{T} P(s_t|s_{t-1}) P(x_t|s_t) \]

(Viterbi algorithm)
How to use Deep Learning?
People imagine ……
This can not be true!
DNN can only take fixed length vectors as input.
What DNN can do is ......

- DNN input: One acoustic feature
- DNN output: Probability of each state

Size of output layer = No. of states

P(a|x_i) P(b|x_i) P(c|x_i) ......

DNN

......

x_i

......
Low rank approximation

- Input layer
- Output layer

$W$: $M \times N$

- $N$ is the size of the last hidden layer
- $M$ is the size of the output layer
- Number of states

$M$ can be large if the outputs are the states of tri-phone.
Low rank approximation

\[ W \approx U V \]

\[ K < M, N \]

Less parameters

Output layer

\[ W \]

Output layer

\[ U, V \]

linear

\[ K \]
How we use deep learning

• There are three ways to use DNN for acoustic modeling
  • Way 1. Tandem
  • Way 2. DNN-HMM hybrid
  • Way 3. End-to-end

Efforts for exploiting deep learning
How to use Deep Learning?

Way 1: Tandem
Way 1: Tandem system

Size of output layer = No. of states

Last hidden layer or bottleneck layer are also possible.
How to use Deep Learning?

Way 2: DNN-HMM hybrid
Way 2: DNN-HMM Hybrid

\[ \widetilde{W} = \arg \max_W P(W | X) = \arg \max_W P(X | W)P(W) \]

\[ P(X | W) \approx \max_{s_1 \cdots s_T} \prod_{t=1}^{T} P(s_t | s_{t-1}) P(x_t | s_t) \]

From DNN

\[ P(x_t | s_t) = \frac{P(x_t, s_t)}{P(s_t)} \]

\[ = \frac{P(s_t | x_t)P(x_t)}{P(s_t)} \]

Count from training data
Way 2: DNN-HMM Hybrid

\[
P(X|W) \approx \max_{s_1 \cdots s_T} \prod_{t=1}^{T} P(s_t|s_{t-1}) P(x_t|s_t)
\]

From original HMM

This assembled vehicle works .......
Way 2: DNN-HMM Hybrid

• **Sequential Training**

\[
\hat{W} = \arg \max_{W} P(X|W; \theta)P(W)
\]

Given training data \((X_1, \hat{W}_1), (X_2, \hat{W}_2), \ldots (X_r, \hat{W}_r), \ldots\)

Find-tune the DNN parameters \(\theta\) such that

\[
P(X_r|\hat{W}_r; \theta)P(\hat{W}_r) \quad \text{increase}
\]

\[
P(X_r|W; \theta)P(W) \quad \text{decrease}
\]

\((W\) is any word sequence different from \(\hat{W}_r\))
How to use Deep Learning?

Way 3: End-to-end
Way 3: End-to-end - Character

Input: acoustic features (spectrograms)

Output: characters (and space) + null (~)

No phoneme and lexicon (No OOV problem)

Way 3: End-to-end – Word?

DNN

Output layer
Size = 200 times of acoustic features
Input layer
padding with zero!

~50k words in the lexicon

Use other systems to get the word boundaries

Why Deep Learning?
Deeper is better?

- Word error rate (WER)

```
<table>
<thead>
<tr>
<th>LxN</th>
<th>DBN-PT (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1×2k</td>
<td>24.2</td>
</tr>
<tr>
<td>2×2k</td>
<td>20.4</td>
</tr>
<tr>
<td>3×2k</td>
<td>18.4</td>
</tr>
<tr>
<td>4×2k</td>
<td>17.8</td>
</tr>
<tr>
<td>5×2k</td>
<td>17.2</td>
</tr>
<tr>
<td>7×2k</td>
<td>17.1</td>
</tr>
</tbody>
</table>
```

Deeper is Better

Deeper is better?

• Word error rate (WER)

For a fixed number of parameters, a deep model is clearly better than the shallow one.

<table>
<thead>
<tr>
<th>LxN</th>
<th>DBN-PT (%)</th>
<th>1xN</th>
<th>DBN-PT (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x2k</td>
<td>24.2</td>
<td>1x3,772</td>
<td>22.5</td>
</tr>
<tr>
<td>2x2k</td>
<td>20.4</td>
<td>1x4,634</td>
<td>22.6</td>
</tr>
<tr>
<td>3x2k</td>
<td>18.4</td>
<td>1x16K</td>
<td>22.1</td>
</tr>
<tr>
<td>4x2k</td>
<td>17.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5x2k</td>
<td>17.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7x2k</td>
<td>17.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What does DNN do?

• Speaker normalization is automatically done in DNN

What does DNN do?

• Speaker normalization is automatically done in DNN

What does DNN do?

- In ordinary acoustic models, all the states are modeled independently
  - Not effective way to model human voice

The sound of vowel is only controlled by a few factors.

http://www.ipachart.com/
What does DNN do?


Output of hidden layer reduce to two dimensions

- The lower layers detect the manner of articulation
- All the states share the results from the same set of detectors.
- Use parameters effectively
Speaker Adaptation
Speaker Adaptation

• Speaker adaptation: use different models to recognition the speech of different speakers
  • Collect the audio data of each speaker

• A DNN model for each speaker
  • Challenge: limited data for training
    • Not enough data for directly training a DNN model
    • Not enough data for just fine-tune a speaker independent DNN model
Categories of Methods

Conservative training
- Re-train the whole DNN with some constraints

Transformation methods
- Only train the parameter of one layer

Speaker-aware Training
- Do not really change the DNN parameters

Need less training data
Conservative Training

Input layer

Output layer

audio data of many speakers

parameter close

output close

A little data from target speaker

initialization
Transformation methods

**Add an extra layer**

- Input layer
- Layer i
- Layer i+1
- Output layer

**Fix all the other parameters**

- Input layer
- Layer i
- Layer i+1
- Output layer

A little data from target speaker
Transformation methods

- Add the extra layer between the input and first layer
- With splicing
Transformation methods

• SVD bottleneck adaptation

\[ M = U V^T \]

Output layer

\[ W_a \]

\[ K \]

K is usually small
Speaker-aware Training

Can also be noise-aware, devise aware training, ......

Lots of mismatched data

Data of Speaker 1

Data of Speaker 2

Data of Speaker 3

Speaker information

Fixed length low dimension vectors

Text transcription is not needed for extracting the vectors.
Speaker-aware Training

Training data:

- Speaker 1: Acoustic features are appended with speaker information features
- Speaker 2: Acoustic features are appended with speaker information features

Testing data: All the speaker use the same DNN model

Different speaker augmented by different features
Multi-task Learning
Multitask Learning

- The multi-layer structure makes DNN suitable for multitask learning
Multitask Learning - Multilingual

Human languages share some common characteristics.

- states of French
- states of German
- states of Spanish
- states of Italian
- states of Mandarin

acoustic features
Multitask Learning - Multilingual

Multitask Learning
- Different units

A = state
B = phoneme
A = state
B = gender
A = state
B = grapheme
(character)

acoustic features
Deep Learning for Acoustic Modeling

New acoustic features
MFCC

Waveform

DFT

spectrogram

Input of DNN

MFCC

DCT

log

filter bank
Filter-bank Output

Waveform

DFT

spectrogram

Input of DNN

Kind of standard now

log

filter bank
5% relative improvement over filterbank output

Spectrogram

Log-Mel Filter Bank Features

Learned Filter Bank Features
People tried, but not better than spectrogram yet


Still need to take Signal & Systems ...... 😊

If success, no Signal & Systems 😊
Figure 1: Four rows from the first layer weight matrix trained on raw time signal. The time range corresponds to 17 frames of 10 ms (17 \cdot 10ms \cdot 16kHz = 2720)
Convolutional Neural Network (CNN)
CNN

- Speech can be treated as images
CNN

Probabilities of states

Replace DNN by CNN

Image
CNN

Applications in Acoustic Signal Processing
DNN for Speech Enhancement

- Demo for speech enhancement: [http://home.ustc.edu.cn/~xuyong62/demo/SE_DNN.html](http://home.ustc.edu.cn/~xuyong62/demo/SE_DNN.html)

- Clean Speech for mobile communication or speech recognition

- Noisy Speech

- DNN
DNN for Voice Conversion

Concluding Remarks
Concluding Remarks

• Conventional Speech Recognition
• How to use Deep Learning in acoustic modeling?
• Why Deep Learning?
• Speaker Adaptation
• Multi-task Deep Learning
• New acoustic features
• Convolutional Neural Network (CNN)
• Applications in Acoustic Signal Processing
Thank you for your attention!
More Researches related to Speech

Find the lectures related to “deep learning”

Spoken Content Retrieval

Speech Recognition

Speech Summarization

I would like to leave Taipei on November 2nd

Computer Assisted Language Learning

Information Extraction

Dialogue

Hi
Hello

lecture recordings

Summary

core techniques