Improved TDNNs using Deep Kernels and Frequency Dependent Grid-RNNs

Florian Kreyssig, Chao Zhang and Phil Woodland

April 18, 2017

Cambridge University Engineering Department
Overview

- Introduction
- Models
  - Baseline TDNN
  - Deep Kernels
  - Frequency-Dependent Grid-RNN
  - Frequency-Dependent CNN (for comparison)
- Experimental Setup (MGB3 English)
- Experimental Results
- Conclusions
Neural Network Depth

- deepening Neural Networks often yields improved performance
- structure of the TDNN restricts its depth
- we deepen the TDNN by exchanging each Kernel of the temporal convolution through a deeper structure

Frequency Dependent Grid-RNNs

- recently 2D-LSTM designs were shown to improve acoustic modelling
- we propose an efficient 2D-RNN design with frequency dependent parameters that as the front-end to a TDNN
Time-Delay Neural Networks (TDNNs) [1]

- consists of FC layers repeated at different time-steps
- parameters are shared across time
- incorporates that the same feature can occur at any time-step
- similar to 1-D (temporal) CNNs
- modern versions use shifts of more than one frame
- version from [2] is used in this work


• current TDNNs are relatively shallow since deep TDNNs need larger input contexts
• TDNN design does not incorporate the structure of the frequency domain
• this work deepens the TDNN, by deepening each convolutional kernel
• spectro-temporal variations will be modelled using a 2D-RNN as a front-end to the TDNN
• both alterations can be combined
Deep Kernels

- replace each convolution kernel in a TDNN with a Deep Kernel
- parameters are still shared across time-domain
- Double Kernel consists of two FC layers
- Resnet Kernel consists of FC layer followed by two further FC layers bypassed with a residual connection
- linear activation function is needed since output range of $\sigma(\cdot)$ is positive

Figure: Darker blocks are FC layers with $\sigma(\cdot)$ activation function. The white block denotes an FC layer with linear activation function.
Frequency-Dependent Grid-RNNs

- 2D-LSTM architectures have shown promising results [3,4]
- LSTMs are unfolded along both the time- and frequency axis
- allows units to influence each other within the same layer
- unfolding for one time-step at a time is expensive
- we exploit TDNN structure and unfold for 7 time-steps (time-bins)
- features at low and high end of the frequency scale are different
- translational weight sharing along frequency axis is sub-optimal


Frequency-Dependent Grid-RNNs
Frequency-Dependent Grid-RNNs

\[ h_{t,k}^l = W_{F1}^l h_{t,k}^F + W_{F2}^l h_{t,k-1}^F + V_i^l h_{t-1,k}^l + b^l \] Combination Matrix

\[ h_{t,k}^F = \sigma \left( W_{(k)}^F x_{t,k} + V_{(k)}^F h_{t,k-1}^l + b_{(k)}^F \right) \] FD-RNN k

- \( x_{t,k} \) is the input at time step \( k \) and frequency step \( k \)
- linear activation in Combination Matrix for better information flow
**Frequency-Dependent Grid-RNNs**

- architecture separates information flow and feature extraction
- one or both axes can be reversed to yield bi-directional FD-RNN
- 5 frequency bins and 7 time bins for easy combination with TDNN
- frequency bins have separate weights (note colours)
- ’FD-RNN 5’ (blue) is followed by the TDNN

<table>
<thead>
<tr>
<th>Time</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-13,-9]</td>
<td>[-10,-6]</td>
</tr>
<tr>
<td>[1,10]</td>
<td>[9,18]</td>
</tr>
</tbody>
</table>

![Diagram of Frequency-Dependent Grid-RNNs](image)
Frequency-Dependent CNN (for comparison)

- the 7 time bins of the TDNN have width 5
- split frequency axis into 7 overlapping frequency bins
- each time-frequency bin is convolved with a set of 5x5 filters
- separate set of filters for each frequency bin
- output is 6x1 for each filter within a time-frequency bin
- reduced to 3x1 via maxpooling
- output of the convolutions within a time bin are passed to the TDNN
Experimental Setup

Data
- 55h and 275h from English Multi-Genre Broadcast (MGB) Challenge 3
- A trigram word level LM with a 63k word dictionary
- dev17b test set contains 5.5h data with reference segmentation

Systems
- All experiments were conducted by extending HTK 3.5
- 40-dim log-Mel filter bank features were used, with $\Delta$ for LSTMP
- number of parameters was kept constant by adjusting layer-sizes
- trained using cross-entropy criterion
- initialized using discriminative pre-training
- evaluation used confusion network decoding
Results 55h: Comparing the three Kernels

- Deep Kernels yield significant improvement

<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>WER</th>
<th>WERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ST^{55h}_1$</td>
<td>TDNN</td>
<td>32.7</td>
<td>–</td>
</tr>
<tr>
<td>$DT^{55h}_1$</td>
<td>Double-TDNN</td>
<td>31.5</td>
<td>3.7%</td>
</tr>
<tr>
<td>$RT^{55h}_1$</td>
<td>ResNet-TDNN</td>
<td>30.5</td>
<td>6.7%</td>
</tr>
</tbody>
</table>

Diagram:
- Standard Kernel
- Double Kernel
- ResNet Kernel

Operations:
- Linear add
- Sigmoid
appending FC-Layers also yields improvement
- can be combined with ResNet-Kernel
- gains from ResNet-Kernels

<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>WER</th>
<th>WERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST$^{55h}_1$</td>
<td>TDNN</td>
<td>32.7</td>
<td>–</td>
</tr>
<tr>
<td>RT$^{55h}_1$</td>
<td>ResNet-TDNN</td>
<td>30.5</td>
<td>6.7%</td>
</tr>
<tr>
<td>ST$^{55h}_2$</td>
<td>TDNN + 1 FC</td>
<td>31.9</td>
<td>2.4%</td>
</tr>
<tr>
<td>ST$^{55h}_3$</td>
<td>TDNN + 2 FC</td>
<td>30.9</td>
<td>5.5%</td>
</tr>
<tr>
<td>ST$^{55h}_4$</td>
<td>TDNN + 3 FC</td>
<td>30.5</td>
<td>6.7%</td>
</tr>
<tr>
<td>RT$^{55h}_2$</td>
<td>ResNet-TDNN + 3 FC</td>
<td>29.8</td>
<td>8.9%</td>
</tr>
</tbody>
</table>
Experimental Results 55h: Combination with Grid-RNN

- Frequency-dependent parameters are important.
- Bi-directional model further improves results.
- Bi-directional FD-Grid-RNN outperforms frequency-dependent CNN.

<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>WER</th>
<th>WERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST$_{55h}^1$</td>
<td>TDNN</td>
<td>32.7</td>
<td>–</td>
</tr>
<tr>
<td>RT$_{55h}^1$</td>
<td>ResNet-TDNN</td>
<td>30.5</td>
<td>6.7%</td>
</tr>
<tr>
<td>RC$_{55h}^1$</td>
<td>FD-CNN-ResNet-TDNN</td>
<td>29.9</td>
<td>8.6%</td>
</tr>
<tr>
<td>RG$_{55h}^1$</td>
<td>Grid-RNN-ResNet-TDNN</td>
<td>30.1</td>
<td>8.0%</td>
</tr>
<tr>
<td>RG$_{55h}^2$</td>
<td>FD-Grid-RNN-ResNet-TDNN</td>
<td>29.6</td>
<td>9.5%</td>
</tr>
<tr>
<td>RG$_{55h}^3$</td>
<td>BD-FD-Grid-RNN-ResNet-TDNN</td>
<td>29.0</td>
<td>11.3%</td>
</tr>
<tr>
<td>L$_{55h}^1$</td>
<td>2L-LSTMP</td>
<td>30.6</td>
<td>6.4%</td>
</tr>
</tbody>
</table>
Experimental Results 275h

- alterations also give large improvements for the larger dataset
- ResNet-Kernel is more effective than appending FC layers on 275h dataset in comparison to 55h dataset

<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>WER</th>
<th>WERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST\textsubscript{275h}\textsubscript{1}</td>
<td>TDNN</td>
<td>26.7</td>
<td>–</td>
</tr>
<tr>
<td>ST\textsubscript{275h}\textsubscript{4}</td>
<td>TDNN + 3 FC</td>
<td>25.7</td>
<td>3.7%</td>
</tr>
<tr>
<td>RT\textsubscript{275h}\textsubscript{1}</td>
<td>ResNet-TDNN</td>
<td>25.0</td>
<td>6.4%</td>
</tr>
<tr>
<td>RT\textsubscript{275h}\textsubscript{2}</td>
<td>ResNet-TDNN + 3 FC</td>
<td>24.7</td>
<td>7.5%</td>
</tr>
<tr>
<td>RG\textsubscript{275h}\textsubscript{3}</td>
<td>BD-FD-Grid-RNN-ResNet-TDNN</td>
<td>24.3</td>
<td>9.0%</td>
</tr>
<tr>
<td>L\textsubscript{275h}\textsubscript{1}</td>
<td>2L-LSTMP</td>
<td>25.6</td>
<td>4.1%</td>
</tr>
</tbody>
</table>
Conclusions

- replacing convolutional kernels in a TDNN with deeper structures improves acoustic modelling (6.4% WERR)
- the best deep kernel consists of three FC layers with a ResNet connection from the output of the first the output of the third
- 2D-RNNs can be used as front-end to TDNN to effectively model spectro-temporal variations
- 2D-RNN design need not rely on LSTMs
- parameters of the 2D-RNNs should be frequency dependent
- the alterations are complimentary (9.0% WERR)
Thanks for listening!