A Study of Comparing Deep Long Short-Term Memory RNN Models for Speech Recognition

Wei-Ning Hsu, Yu Zhang, James Glass

Spoken Language Systems Group
Computer Science and Artificial Intelligence Laboratory (CSAIL)
Massachusetts Institute of Technology (MIT)
Cambridge, MA, USA

2016.10.18
Outline

▶ Background
▶ Motivation
▶ Challenges
▶ Proposed Solutions
▶ Experiment Setup
▶ Results
▶ Conclusions
Background-1

- An automatic speech recognition system (ASR) can be decomposed into three components
  - Acoustic Model (AM)
  - Lexicon
  - Language Model (LM)
- The acoustic model predicts which phonetic unit is being pronounced
  - each context-dependent phoneme (triphone) is characterized by a hidden Markov model (HMM)
Background-2

► How to model the emission probability of tied HMM-states (senones)?
  – Gaussian Mixture Models (GMMs)
  – Neural Network (NNs)

► Which neural network model to use?
  – Feed-Forward Neural Network (DNN)
  – Convolutional Neural Network (CNN)
  – Recurrent Neural Network (RNN)
Recurrent Neural Networks (RNNs)

Forward operation:

\[
x_t^l = [h_t^{l-1}; h_{t-1}^l]
\]

\[
h_t^l = \tanh(W^l x + b^l)
\]
Motivation

- RNN has shown great success for acoustic modeling
  - utilize dynamically changing contextual windows
  - models the temporal relationships at different time granularities\(^1\)
- Deeper models have shown improvement in other fields
  - Computer Vision: VGG19\(^2\) (19 layers)
  - Natural Language Processing: ConvNet\(^3\) (29 layers)
- Can we build deeper RNN models for ASR?


Issues with Deeper Models

► Vanishing/Exploding Gradient
  – optimize the network using gradient-based methods
    ► obtain the gradient by differentiating the error signal with respect to the parameters
  – error signals back-propagate through many layers and nonlinear transformations $\rightarrow$ exponential decay/growth
An Example Remedy for the Gradient Problem

- The victim: Recurrent Neural Network (RNN)
  - pass information to the same layer at the next time step
  - deep in time, error signals back propagate through many time steps
- The remedy: Long Short-Term Memory (LSTM) RNNs
  - cells as internal memory states
  - introduce gated linear dependence between temporally adjacent cells
Visualizing the Gradient Problem in Deep LSTM

- Blue line denotes the information flow along the depth dimension
- Twice non-linearities at each layer

Highway LSTM

\[
\begin{align*}
x^l_t &= [h^l_{t-1}; h^l_{t-1}] \\
i^l_t &= \sigma(W^l_i x^l_t + U^l_i c^l_{t-1} + b^l_i) \\
f^l_t &= \sigma(W^l_f x^l_t + U^l_f c^l_{t-1} + b^l_f) \\
\hat{c}^l_t &= \tanh(W^l_c x^l_t + U^l_c c^l_{t-1} + b^l_c) \\
c^l_t &= f^l_t \odot c^l_{t-1} + i^l_t \odot \hat{c}^l_t \\
o^l_t &= \sigma(W^l_o x^l_t + U^l_o c^l_t + b^l_o) \\
h^l_t &= o^l_t \odot \tanh(c^l_t)
\end{align*}
\]
Remedy 1 – Highway Connections (Depth Gate)

- Introduce gated linear dependence between cells in adjacent layers
- Smoothly vary the behavior from short circuit to stacked LSTM

Highway LSTM

\[
d^l_t = f(x^l_t, c^l_{t-1}, c^{l-1}_t; W^l_d)
\]

\[
c^l_t = i^l_t \odot \tilde{c}^l_t + f^l_t \odot c^{l-1}_t + d^l_t \odot c^{l-1}_t
\]
Remedy 2 – Shortcut Connections (ResNet)

- Introduce shortcut connections from input of an LSTM cell to its output
- Upper layer learns the residual function

Residual LSTM

\[ h_t^l = o_t^l \odot \tanh(c_t^l) + h_{t-1}^{l-1} \]
Remedy 3 – Grid LSTM

- Arrange LSTM blocks into multidimensional grids such that each grid contains one set of LSTM blocks for each dimension
- Introduce per-dimension gated linear dependencies between adjacent cell states

Grid LSTM

\[
\begin{align*}
x_{t,l} &= [h_{t,l-1}^D; h_{t-1,l}^T] \\
(h_{t,l}^T, c_{t,l}^T) &= \text{TIME-LSTM}(x_{t,l}, c_{t-1,l}^T; \Theta^T) \\
(h_{t,l}^D, c_{t,l}^D) &= \text{DEPTH-LSTM}(x_{t,l}, c_{t,l-1}^D; \Theta^D) \\
\text{Initialization:} & \quad c_{t,0}^D = V h_{t,0}^D
\end{align*}
\]
Remedy 3.1 – Prioritized Grid LSTM

- The cell output from **Time-LSTM** is not being utilized for classification of the current time step
- Input to **Depth-LSTM** is updated after **Time-LSTM** of the same grid is processed

Prioritized Grid LSTM

\[
x_{t,l}^T = [h_{t,l-1}^D; h_{t-1,l}^T]
\]

\[
(h_{t,l}^T, c_{t,l}^T) = \text{TIME-LSTM}(x_{t,l}^T, c_{t-1,l}^T, \Theta^T)
\]

\[
x_{t,l}^D = [h_{t,l-1}^D; h_{t,l}^T]
\]

\[
(h_{t,l}^D, c_{t,l}^D) = \text{DEPTH-LSTM}(x_{t,l}^D, c_{t,l-1}^D, \Theta^D)
\]
Experiment Setup

- Dataset
  - AMI English meetings (100hr)
  - HKUST Mandarin telephone speech (150hr)
  - Switchboard English telephone speech (300hr)
  - GALE Chinese broadcast conversation/news (500hr)
  - MGB Arabic broadcast programs (1200hr)

- Model
  - Input: 80-d log Mel filterbank + 3-d pitch features
  - Output: forced alignment from ML-criterion context-dependent speaker-adapted GMM
  - Training: cross entropy criterion, back-propagation through time unrolling 20 frames, SGD with momentum (from the second epoch)
Experiment 1
Effectiveness of Prioritizing Depth Dimension on HKUST and GALE

<table>
<thead>
<tr>
<th>Model</th>
<th>#layers</th>
<th>HKUST</th>
<th>GALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>3</td>
<td>33.29</td>
<td>23.96</td>
</tr>
<tr>
<td>HLSTM</td>
<td>3</td>
<td>32.86</td>
<td>23.33</td>
</tr>
<tr>
<td>npGLSTM</td>
<td>3</td>
<td>32.32</td>
<td>22.80</td>
</tr>
<tr>
<td>pGLSTM</td>
<td>3</td>
<td>32.06</td>
<td>22.54</td>
</tr>
</tbody>
</table>
## Experiment 2

Comparing Different Remedies for Deep LSTMs on AMI

<table>
<thead>
<tr>
<th>Model</th>
<th>#layers</th>
<th>#params</th>
<th>with overlap</th>
<th>no overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>3</td>
<td>12M</td>
<td>50.7</td>
<td>41.7</td>
</tr>
<tr>
<td>LSTM</td>
<td>8</td>
<td>36M</td>
<td>52.6</td>
<td>43.8</td>
</tr>
<tr>
<td>HLSTM</td>
<td>3</td>
<td>14M</td>
<td>50.4</td>
<td>41.2</td>
</tr>
<tr>
<td>HLSTM</td>
<td>8</td>
<td>40M</td>
<td>50.7</td>
<td>41.3</td>
</tr>
<tr>
<td>HLSTM</td>
<td>16</td>
<td>82M</td>
<td>50.7</td>
<td>41.2</td>
</tr>
<tr>
<td>RLSTM</td>
<td>3</td>
<td>12M</td>
<td>51.3</td>
<td>42.0</td>
</tr>
<tr>
<td>RLSTM</td>
<td>8</td>
<td>36M</td>
<td>50.5</td>
<td>40.8</td>
</tr>
<tr>
<td>RLSTM</td>
<td>16</td>
<td>74M</td>
<td>49.9</td>
<td>40.4</td>
</tr>
<tr>
<td>pGLSTM</td>
<td>3</td>
<td>25M</td>
<td>49.8</td>
<td>40.5</td>
</tr>
<tr>
<td>pGLSTM</td>
<td>8</td>
<td>72M</td>
<td>49.0</td>
<td>39.6</td>
</tr>
</tbody>
</table>
## Experiment 3
Verifying Results on More Datasets

<table>
<thead>
<tr>
<th>Model</th>
<th>#L</th>
<th>AMI</th>
<th>HKUST</th>
<th>SWB</th>
<th>GALE</th>
<th>MGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>BHLSTM</td>
<td>3</td>
<td>48.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M-LSTM</td>
<td>3</td>
<td>-</td>
<td>33.89</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HCLDNN</td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>22.41</td>
<td>-</td>
</tr>
<tr>
<td>LSTM</td>
<td>3</td>
<td>50.7</td>
<td>33.29</td>
<td>20.5</td>
<td>23.96</td>
<td>23.56</td>
</tr>
<tr>
<td>HLSTM</td>
<td>3</td>
<td>50.4</td>
<td>32.86</td>
<td>20.6</td>
<td>23.33</td>
<td>23.32</td>
</tr>
<tr>
<td>HLSTM</td>
<td>5</td>
<td>50.7</td>
<td>32.40</td>
<td>19.8</td>
<td>22.63</td>
<td>23.12</td>
</tr>
<tr>
<td>pGLSTM</td>
<td>3</td>
<td>49.8</td>
<td>32.06</td>
<td>19.4</td>
<td>22.54</td>
<td>22.36</td>
</tr>
<tr>
<td>pGLSTM</td>
<td>5</td>
<td>48.6</td>
<td>31.36</td>
<td>19.3</td>
<td>22.33</td>
<td>22.18</td>
</tr>
</tbody>
</table>
Conclusions

1. Prioritizing the depth dimension is essential for achieving better performance,

2. The prioritized grid LSTM model outperforms two alternative designs for deep LSTM models,

3. Results are consistent on five datasets
Thank you!