Long Short-Term Memory over Recursive Structures

Xiaodan Zhu
National Research Council Canada (NRC), Ottawa

Parinaz Sobhani
EECS, University of Ottawa

Hongyu Guo
National Research Council Canada (NRC), Ottawa
Linear-chain LSTM

• In linear-chain LSTM, history is summarized and encoded in *memory cells* in a *sequential* fashion.
Recursive LSTM

• Recursion and the structures it forms are common in different modalities, e.g., trees [Socher, ’12; ’13].

• While linear-chain LSTM can be used to model such problems, we take a different view point.
Recursive LSTM

- We propose to extend LSTM to recursive (tree here) structures.
- We aim to explore a good way to consider invariants and long-distance interplays over given structures.
  - E.g., the distance/relationship between $n_1$ and $n_2$ are invariant if node $p$ varies (e.g., as a node of noun or a subtree of a longer phrase).
  - Such a model is interesting to us also because it recursively summarizes history over structure constituents.
The Memory Blocks

LSTM

S-LSTM (Our model)
S-LSTM: Forward Propagation

\[ i_t = \sigma(W_{hi}^L h_{t-1}^L + W_{hi}^R h_{t-1}^R + W_{ci}^L c_{t-1}^L + W_{ci}^R c_{t-1}^R + b_i) \]  
\[ f_t^L = \sigma(W_{hf_l}^L h_{t-1}^L + W_{hf_l}^R h_{t-1}^R + W_{cf_l}^L c_{t-1}^L + W_{cf_l}^R c_{t-1}^R + b_{f_l}) \]  
\[ f_t^R = \sigma(W_{hf_r}^L h_{t-1}^L + W_{hf_r}^R h_{t-1}^R + W_{cf_r}^L c_{t-1}^L + W_{cf_r}^R c_{t-1}^R + b_{f_r}) \]  
\[ x_t = W_{hx}^L h_{t-1}^L + W_{hx}^R h_{t-1}^R + b_x \]  
\[ c_t = f_t^L c_{t-1}^L + f_t^R c_{t-1}^R + i_t \tanh(x_t) \]  
\[ o_t = \sigma(W_{ho}^L h_{t-1}^L + W_{ho}^R h_{t-1}^R + W_{co} c_t + b_o) \]  
\[ h_t = o_t \tanh(c_t) \]
S-LSTM: Backpropagation

\[ e_t^h = \frac{\partial O}{\partial h_t} \]  \hspace{1cm} (8)

\[ \delta_t^o = e_t^h \otimes \tanh(c_t) \otimes f'(o_t) \]  \hspace{1cm} (9)

\[ \delta_t^{f_l} = e_t^c \otimes c_{t-1}^L \otimes f'(f_t^L) \]  \hspace{1cm} (10)

\[ \delta_t^{f_r} = e_t^c \otimes c_{t-1}^R \otimes f'(f_t^R) \]  \hspace{1cm} (11)

\[ \delta_t^i = e_t^c \otimes \tanh(x_t) \otimes f'(i_t) \]  \hspace{1cm} (12)

**Left child:**

\[ e_t^c = e_t^h \otimes o_t \otimes g'(\tanh(c_t)) + e_{t+1}^c \otimes f_{t+1}^L + (W_{ci})^T \delta_{t+1}^i + (W_{cф_l})^T \delta_{t+1}^{f_l} + (W_{cф_r})^T \delta_{t+1}^{f_r} + (W_{co})^T \delta_t^o \]  \hspace{1cm} (13)

**Right child:**

\[ e_t^c = e_t^h \otimes o_t \otimes g'(\tanh(c_t)) + e_{t+1}^c \otimes f_{t+1}^R + (W_{ci})^T \delta_{t+1}^i + (W_{cф_l})^T \delta_{t+1}^{f_l} + (W_{cф_r})^T \delta_{t+1}^{f_r} + (W_{co})^T \delta_t^o \]  \hspace{1cm} (14)

**Handling non-binary trees?**
Experiments
(Sentiment analysis)
Can a machine *fall in love*?
Love:

a (1) : strong affection for another arising out of kinship or personal ties
<maternal love for a child> (2) : attraction based on sexual desire : affection and tenderness felt by lovers (3) : affection based on admiration, benevolence, or common interests <love for his old schoolmates>

... ...

—Merriam-Webster Dictionary
Love:

a (1) : strong affection for another arising out of kinship or personal ties 
<maternal love for a child> (2) : attraction based on sexual desire : affection and tenderness felt by lovers (3) : affection based on admiration, benevolence, or common interests <love for his old schoolmates>

... ...

—Merriam-Webster Dictionary

Love, admiration, satisfaction ...

Anger, fear, hunger ...
Semantics/Sentiment composition
(Figure is borrowed from [Socher et al. ’13])
Semantics/Sentiment composition
(Figure is borrowed from [Socher et al. ’13])
Experiment Set-up

• Data: Stanford Sentiment Treebank
  ◦ Movie reviews
    • # sentences: 8544/1101/2210 (training/dev./test)
    • # phrases: 318582/41447/82600
  ◦ All phrases, including roots (sentences), are manually annotated with sentiment labels.

• Evaluation metric
  ◦ Classification accuracy (5-category)
Recursive Neural Tensor Network (RNTN)  
[Socher et al., ’13]

\[ p = \tanh \left( \begin{bmatrix} a \\ b \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ b \end{bmatrix} + W \begin{bmatrix} a \\ b \end{bmatrix} \right) \]

\[ p = f \left( \begin{bmatrix} \text{Slices of } \\ \text{Tensor Layer} \end{bmatrix} + \begin{bmatrix} \text{Standard} \\ \text{Layer} \end{bmatrix} \right) \]
Results
(Default setting)

*Table 1.* Performances (accuracies) of different models on the test set of Stanford Sentiment Treebank, at the sentence level (roots) and the phrase level. † shows the performance are statistically significantly better ($p < 0.05$) than the corresponding models.

<table>
<thead>
<tr>
<th>MODELS</th>
<th>ROOTS</th>
<th>PHRASES</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>41.0</td>
<td>67.2</td>
</tr>
<tr>
<td>SVM</td>
<td>40.7</td>
<td>64.3</td>
</tr>
<tr>
<td>RVNN</td>
<td>43.2</td>
<td>79.0</td>
</tr>
<tr>
<td>RNTN</td>
<td>45.7</td>
<td>80.7</td>
</tr>
<tr>
<td>S-LSTM</td>
<td><strong>48.9†</strong></td>
<td><strong>81.9†</strong></td>
</tr>
</tbody>
</table>
Results
( Default setting )

Table 2. Comparison with more models on five-category classification accuracies on the test set of Stanford Sentiment Treebank at the sentence level (roots).

<table>
<thead>
<tr>
<th>MODELS</th>
<th>ROOTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kalchbrenner et al., 2014</strong></td>
<td>48.5</td>
</tr>
<tr>
<td>S-LSTM</td>
<td>48.9</td>
</tr>
<tr>
<td><strong>Irsoy &amp; Cardie, 2014</strong></td>
<td>49.8</td>
</tr>
<tr>
<td>S-LSTM (GloVe-300)</td>
<td>50.1</td>
</tr>
</tbody>
</table>
A Real-life Set-up

• In the default setting, all phrases are manually annotated with sentiment labels, which is often not a real-life set-up.

• Now, we keep only root annotations and root+leaf annotations (imitating the use of lexicons).
Results in a *Real-life* Set-up

*Table 3.* Performances of models trained with only root labels (the first two rows) and models that use both root and leaf labels (the last two rows).

<table>
<thead>
<tr>
<th>Models</th>
<th>Roots</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) RNTN (Root Lbls)</td>
<td>29.1</td>
</tr>
<tr>
<td>(2) S-LSTM (Root Lbls)</td>
<td>43.5†</td>
</tr>
<tr>
<td>(3) RNTN (Root + Leaf Lbls)</td>
<td>34.9</td>
</tr>
<tr>
<td>(4) S-LSTM (Root + Leaf Lbls)</td>
<td>44.1†</td>
</tr>
</tbody>
</table>
Table 4. Performances of models that do not use the given sentence structures. S-LSTM-LR is a degenerated version of S-LSTM that reads input words from left to right, and S-LSTM-RR reads words from right to left.

<table>
<thead>
<tr>
<th>Models</th>
<th>Roots</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-LSTM-LR (Root LbLS)</td>
<td>40.2</td>
</tr>
<tr>
<td>S-LSTM-RR (Root LbLS)</td>
<td>40.3</td>
</tr>
<tr>
<td>S-LSTM (Root LbLS)</td>
<td>43.5†</td>
</tr>
<tr>
<td>S-LSTM-LR (Root + Leaf LbLS)</td>
<td>43.1</td>
</tr>
<tr>
<td>S-LSTM-RR (Root + Leaf LbLS)</td>
<td>43.2</td>
</tr>
<tr>
<td>S-LSTM (Root + Leaf LbLS)</td>
<td>44.1†</td>
</tr>
</tbody>
</table>
Summary

• A tree-structured recursive LSTM model.

• Achieved the state-of-the-art performance on a semantic composition task.

• Explicitly modeling the structures is helpful.

Code is available by email!
Some Related Work

• Related
  ◦ Irsoy & Cardie, NIPS-14;
  ◦ Kalchbrenner et al., ACL-14;
  ◦ Socher et al., EMNLP-13;
  ◦ Hermann et. al, ACL-13

• More related
  ◦ Tai et al. ACL-15
  ◦ Le & Zuidema *SEM-15
Thank you!