NLP
Deep Learning

Long Short-Term Memory Networks (LSTM)
LSTM Motivation

Remember how we update an RNN?

[slides from Catherine Finegan-Dollak]
The Vanishing Gradient Problem

- Deep neural networks use backpropagation.
- Backpropagation uses the chain rule.
- The chain rule multiplies derivatives.
- Often these derivatives are between 0 and 1.
- As the chain gets longer, products get smaller until they disappear.
Or do they explode?

- With gradients larger than 1,
- you encounter the opposite problem
- with products becoming larger and larger
- as the chain becomes longer and longer,
- causing overlarge updates to parameters.

This is the exploding gradient problem.
Vanishing/Exploding Gradients Are Bad.

• If we cannot backpropagate very far through the network, the network cannot learn long-term dependencies.

  • My dog [chase/chases] squirrels. ✓

  vs.

LSTM Solution

- Use memory cell to store information at each time step.
- Use “gates” to control the flow of information through the network.
  - Input gate: protect the current step from irrelevant inputs
  - Output gate: prevent the current step from passing irrelevant outputs to later steps
  - Forget gate: limit information passed from one cell to the next
Transforming RNN to LSTM

\[ u_t = \sigma(W_h h_{t-1} + W_x x_t) \]
Transforming RNN to LSTM
Transforming RNN to LSTM

\[ c_t = f_t \odot c_{t-1} + i_t \odot u_t \]
Transforming RNN to LSTM

\[ c_t = f_t \odot c_{t-1} + i_t \odot u_t \]
Transforming RNN to LSTM

\[ c_t = f_t \odot c_{t-1} + i_t \odot u_t \]
Transforming RNN to LSTM

\[ f_t = \sigma(W_{hf} h_{t-1} + W_{xf} x_t) \]
Transforming RNN to LSTM

\[ i_t = \sigma(W_{hi} h_{t-1} + W_{xi} x_t) \]
Transforming RNN to LSTM

\[ h_t = o_t \odot \tanh c_t \]
LSTM for Sequences

The cat sat
LSTM Applications

- Language identification (Gonzalez-Dominguez et al., 2014)
- Paraphrase detection (Cheng & Kartsaklis, 2015)
- Speech recognition (Graves, Abdel-Rahman, & Hinton, 2013)
- Handwriting recognition (Graves & Schmidhuber, 2009)
- Music composition (Eck & Schmidhuber, 2002) and lyric generation (Potash, Romanov, & Rumshisky, 2015)
- Robot control (Mayer et al., 2008)
- Natural language generation (Wen et al., 2015) (best paper at EMNLP)
- Named entity recognition (Hammerton, 2003)

... and many more
Example: Language Generation

- Wen et al. (2015)
- Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems
Wen et al., 2015 (continued)
<table>
<thead>
<tr>
<th>Method</th>
<th>SF Restaurant</th>
<th>SF Hotel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>ERR(%)</td>
</tr>
<tr>
<td>hdc</td>
<td>0.451</td>
<td>0.0</td>
</tr>
<tr>
<td>kNN</td>
<td>0.602</td>
<td>0.87</td>
</tr>
<tr>
<td>classlm</td>
<td>0.627</td>
<td>8.70</td>
</tr>
<tr>
<td>rnn w/o</td>
<td>0.706</td>
<td>4.15</td>
</tr>
<tr>
<td>lstm w/o</td>
<td>0.714</td>
<td>1.79</td>
</tr>
<tr>
<td>rnn w/</td>
<td>0.710</td>
<td>1.52</td>
</tr>
<tr>
<td>lstm w/</td>
<td>0.717</td>
<td>0.63</td>
</tr>
<tr>
<td>sc-lstm</td>
<td>0.711</td>
<td>0.62</td>
</tr>
<tr>
<td>+deep</td>
<td>0.731</td>
<td>0.46</td>
</tr>
</tbody>
</table>
Related Architectures: GRU

Chung et al. (2014) reports comparable performance to LSTM
Related Architectures: Tree LSTMs

Tai, Socher, Manning 2015
External Links

- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
NLP