Towards Binary-Valued Gates for Robust LSTM Training

Zhuohan Li, Di He, Fei Tian, Wei Chen, Tao Qin, Liwei Wang, Tie-Yan Liu

Peking University & Microsoft Research Asia

ICML | 2018
Long Short-Term Memory (LSTM) RNN

• \( h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, x_t) \)

• \( f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \)
• \( i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \)
• \( g_t = \text{tanh}(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \)
• \( o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \)
• \( c_t = f_t \odot c_{t-1} + i_t \odot g_t \)
• \( h_t = o_t \odot \text{tanh}(c_t) \)

Figure credit to: Christopher Olah, "Understanding LSTM Networks"
Long Short-Term Memory (LSTM) RNN

- $h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, x_t)$
- $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$
- $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$
- $g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$
- $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$
- $c_t = f_t \odot c_{t-1} + i_t \odot g_t$
- $h_t = o_t \odot \tanh(c_t)$

Figure credit to: Christopher Olah, "Understanding LSTM Networks"
Long Short-Term Memory (LSTM) RNN

- \( h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, x_t) \)
- \( f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \)
- \( i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i) \)
- \( g_t = \tanh(W_{xg} x_t + W_{hg} h_{t-1} + b_g) \)
- \( o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \)
- \( c_t = f_t \circ c_{t-1} + i_t \circ g_t \)
- \( h_t = o_t \circ \tanh(c_t) \)

Figure credit to: Christopher Olah, "Understanding LSTM Networks"
Long Short-Term Memory (LSTM) RNN

• $h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, x_t)$

• $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$
• $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$
• $g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$
• $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$
• $c_t = f_t \odot c_{t-1} + i_t \odot g_t$
• $h_t = o_t \odot \tanh(c_t)$

Figure credit to: Christopher Olah, "Understanding LSTM Networks"
Example: Input Gates & Forget Gates

- When the LSTM sees "France", the input gate will open and the LSTM will remember the information.
- At the subsequent timesteps, the forget gates will also be open (take value 1) to keep the information. Finally, the LSTM will use this information to predict word "French".
Example: Input Gates & Forget Gates

• When the LSTM sees "but" and "back", the forget gates should be closed (take value 0) to forget the information of "left"
Histograms of Gate Distributions in LSTM

Based on the gate outputs of the first-layer LSTM in the decoder from 10000 sentence pairs IWSLT14 German→English training sets
Push the gate values to the boundary of range (0, 1)

Well aligns with the original purpose of gates: to get the information in or skip by "opening" or "closing"

Ready for further compression by pushing the activation function to be binarized

Enables better generalization
Ready for Further Compression & Better Generalization

The output falls in the saturation area
Parameters in the gates perturb
Change to the output of the gates will be small
Change to the final loss will also be little

Robust to model compression
Better test performance
Flat region generalizes better

Saturation area of Sigmoid function
Sharpened Sigmoid

- Straight forward idea: **sharpen the Sigmoid function** by using a smaller temperature $\tau < 1$
  \[ f_{W,b}(x) = \sigma((Wx + b)/\tau) = \sigma((W/\tau)x + (b/\tau)) \]
- This is equivalent to **rescale** the weight initialization and the gradient

Harm the **optimization process**

Cannot guarantee the outputs to be **close to the boundary**
Gumbel-Softmax Estimator

• In our special case, we leverage the **Gumbel-Softmax estimator** to estimate the Bernoulli distribution $D_\alpha \sim B(\sigma(\alpha))$ with prob. $\sigma(\alpha)$

• Define

$$G(\alpha, \tau) = \sigma\left(\frac{\alpha + \log U - \log(1 - U)}{\tau}\right),$$

where $U \sim \text{Uniform}(0, 1)$, then the following holds for $\varepsilon \in (0, 1/2)$:

- $P(D_\alpha = 1) - (\tau/4) \log(1/\varepsilon) \leq P(G(\alpha, \tau) \geq 1 - \varepsilon) \leq P(D_\alpha = 1)$
- $P(D_\alpha = 0) - (\tau/4) \log(1/\varepsilon) \leq P(G(\alpha, \tau) \leq \varepsilon) \leq P(D_\alpha = 0)$
Gumbel-Softmax Estimator

Probability density functions of Gumbel-Softmax estimators with different temperature $\tau$

- $\tau = 0$
- $\tau = 1/2$
- $\tau = 1$
- $\tau = 2$
Gumbel-Gate LSTM (G²-LSTM)

- \( h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, x_t) \)
- \( f_t = G(W_{xf} x_t + W_{hf} h_{t-1} + b_f, \tau) \)
- \( i_t = G(W_{xi} x_t + W_{hi} h_{t-1} + b_i, \tau) \)
- \( g_t = \tanh(W_{xg} x_t + W_{hg} h_{t-1} + b_g) \)
- \( o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \)
- \( c_t = f_t \odot c_{t-1} + i_t \odot g_t \)
- \( h_t = o_t \odot \tanh(c_t) \)

In the forward pass during training, we **independently sample** all forget and input gates at each timestep, and update G²-LSTM.

In the backward pass, we use **standard gradient-based method** to update model parameters, since all components are differentiable.
Experiments

• Language Modeling
  • Penn Treebank

• Machine Translation
  • IWSLT'14 German→English
  • WMT'14 English→German
Sensitivity Analysis

• Compress the gate-related parameters to show the robustness of our learned models

- **Low-precision compression**
  - Reduce the support set of the parameters by
    \[
    \text{round}_r(x) = \text{round}(x / r) \cdot r
    \]
  - Further clip the rounded value to a fixed range using
    \[
    \text{clip}_c(x) = \text{clip}(x, -c, c)
    \]

- **Low-rank compression**
  - Compress the parameter matrices by **singular value decomposition** (SVD)
  - Reduce the model size and lead to fast matrix multiplication
### Experimental Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Original</th>
<th>Round</th>
<th>Round &amp; Clip</th>
<th>SVD</th>
<th>SVD+</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Penn Treebank (Perplexity)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>52.8</td>
<td>53.2 (+0.4)</td>
<td>53.6 (+0.8)</td>
<td>56.6 (+3.8)</td>
<td>65.5 (+12.7)</td>
</tr>
<tr>
<td>Sharpened Sigmoid</td>
<td>53.2</td>
<td>53.5 (+0.3)</td>
<td>53.6 (+0.4)</td>
<td>54.6 (+1.4)</td>
<td>60.0 (+6.8)</td>
</tr>
<tr>
<td>G&lt;sup&gt;2&lt;/sup&gt;-LSTM</td>
<td><strong>52.1</strong></td>
<td><strong>52.2 (+0.1)</strong></td>
<td><strong>52.8 (+0.7)</strong></td>
<td><strong>53.3 (+1.2)</strong></td>
<td><strong>56.0 (+3.9)</strong></td>
</tr>
<tr>
<td><strong>IWSLT’14 German→English (BLEU)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>31.00</td>
<td>28.65 (-2.35)</td>
<td>21.97 (-9.03)</td>
<td>30.52 (-0.48)</td>
<td>29.56 (-1.44)</td>
</tr>
<tr>
<td>Sharpened Sigmoid</td>
<td>29.73</td>
<td>27.08 (-2.65)</td>
<td>25.14 (-4.59)</td>
<td>29.17 (-0.53)</td>
<td>28.82 (-0.91)</td>
</tr>
<tr>
<td>G&lt;sup&gt;2&lt;/sup&gt;-LSTM</td>
<td><strong>31.95</strong></td>
<td><strong>31.44 (-0.51)</strong></td>
<td><strong>31.44 (-0.51)</strong></td>
<td><strong>31.62 (-0.33)</strong></td>
<td><strong>31.28 (-0.67)</strong></td>
</tr>
<tr>
<td><strong>WMT’14 English→German (BLEU)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>21.89</td>
<td>16.22 (-5.67)</td>
<td>16.03 (-5.86)</td>
<td>21.15 (-0.74)</td>
<td>19.99 (-1.90)</td>
</tr>
<tr>
<td>Sharpened Sigmoid</td>
<td>21.64</td>
<td>16.85 (-4.79)</td>
<td>16.72 (-4.92)</td>
<td>20.98 (-0.66)</td>
<td>19.87 (-1.77)</td>
</tr>
<tr>
<td>G&lt;sup&gt;2&lt;/sup&gt;-LSTM</td>
<td><strong>22.43</strong></td>
<td><strong>20.15 (-2.28)</strong></td>
<td><strong>20.29 (-2.14)</strong></td>
<td><strong>22.16 (-0.27)</strong></td>
<td><strong>21.84 (-0.51)</strong></td>
</tr>
</tbody>
</table>
Histograms of Gate Distributions in G²-LSTM

Based on the gate outputs of the first-layer G²-LSTM in the decoder from the same 10000 sentence pairs IWSLT14 German→English training sets
Visualization of Average Gate Values
Summary

• A new training algorithm for LSTM by leveraging the recently developed Gumbel-Softmax estimator

• Push the values of the input and forget gates to 0 or 1, leading to robust LSTM models

• Experiments on language modeling and machine translation demonstrated the effectiveness of the proposed training algorithm
Thanks!

Poster #63

Contact:

Zhuohan Li (lizhuohan@pku.edu.cn)
Tao Qin (taoqin@microsoft.com)

Zhuohan is applying for a Ph.D. in Fall 2018! Please contact if you are interested!