Neural Speed Reading via Skim-RNN

Minjoon Seo\textsuperscript{1,2}, Sewon Min\textsuperscript{3}, Ali Farhadi\textsuperscript{4,5}, Hannaneh Hajishirzi\textsuperscript{1}

University of Washington\textsuperscript{1}, NAVER Clova\textsuperscript{2}, Seoul National University\textsuperscript{3}, Allen Institute for AI\textsuperscript{4}, XNOR.AI\textsuperscript{5}

ICLR 2018 @ Vancouver, Canada - April 30 (Mon)

**Motivation**

RNN on CPUs
- RNNs are slow on CPUs/GPUs
- CPUs are often more desirable than GPUs

Human Speed Reading
- **Skim** unimportant words and **fully read** important words

Our contributions
- Dynamically decide which one to use between big and small RNN with shared hidden state
- Same interface as standard RNN → Easily Replaceable
- Computational Advantage over standard RNN on CPUs with comparable/better accuracy

**Model**

Consists of two RNNs, sharing hidden state
- **Big RNN** (hidden size of d)
  - updates the entire hidden state. $O(d^2)$
- **Small RNN** (hidden size of $d'$, where $d>d'$)
  - updates a small portion of the hidden state. $O(d'd')$

$x_t$: Input state at $t$
$h_t$: Hidden state at $t$
$q_t$: Random variable for skim decision at $t$

Model update equations:

\[
\begin{align*}
 p_t &= \text{softmax}(\alpha(x_t, h_{t-1})) \\
 q_t &= \text{Multinomial}(p_t) \\
 h_t &= \begin{cases} 
 f(x_t, h_{t-1}), & \text{if } q_t = 1 \\
 f'(x_t, h_{t-1}; f'; h_{t-1}[d' + 1 : d]), & \text{if } q_t = 2
\end{cases} \\
\text{if } q_t &= 1
\end{align*}
\]

**Training**

\[
L'(\theta; Q) = L(\theta; Q) + \frac{1}{T} \sum_{t=1}^{T} (\log(\text{Pr}(q_t = 2)))
\]

(y: hyperparameter, encourages skimming)

**Goal:** minimize $E[L'(\theta)]$ = $D(\theta; Q)$ over sample $Q = \{q_t, \ldots, q_T\}$

**Gumbel-Softmax**
- Biased but low variance, good empirical results
- Reparameterize $h_t = \frac{r_t}{1 + \beta} h_t + r_t \gamma$

\[
\gamma \sim \text{Gumbel}(0,1), \quad \tau: \text{hyperparameter}
\]

- Slowly anneal (decrease $\tau$), making the distribution more discrete to allow differentiation with stochasticity

**Related Work**

LSTM-Jump (Yu et al., 2017)
- Skip rather than skim
- No outputs for skipped time steps

Variable-Computation RNN (VCRNN) (Jernite et al., 2017)
- Use 1 RNN, and controls # of hidden units to update by making a d-way decision
- Required to be invariant of # of hidden units being updated
- Computing loss is more tractable ($d'$ vs $2d'$)

**Experiments**

**Text Classification (w/ LSTM)**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SST</th>
<th>Rotten Tomatoes</th>
<th>IMdb</th>
<th>AGNews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>86.4</td>
<td>82.5</td>
<td>91.1</td>
<td>93.5</td>
</tr>
<tr>
<td>LSTM-Jump</td>
<td>79.3 / 1.6x Sp</td>
<td>89.4 / 1.6x Sp</td>
<td>89.3 / 1.1x Sp</td>
<td></td>
</tr>
<tr>
<td>VCRNN</td>
<td>81.9 / 2.6x FLOP-R</td>
<td>81.4 / 1.6x Sp</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Skim-RNN</td>
<td>86.4 / 3.0x FLOP-R</td>
<td>84.2 / 1.3x Sp</td>
<td>91.2 / 2.3x Sp</td>
<td>93.6 / 1.0x Sp</td>
</tr>
</tbody>
</table>

▲ Accuracy, Floating point operation reduction (FLOP-R) and Speed-up (Sp). Skim-RNN achieves comparable to/better than regular RNN and related work, with up to 3.0x FLOP-R.

**Question Answering (SQuAD) (w/ LSTM+Attention)**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>F1</th>
<th>Exact Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>75.5</td>
<td>67.0</td>
</tr>
<tr>
<td>Skim-RNN</td>
<td>75.0</td>
<td>66.0</td>
</tr>
</tbody>
</table>

▲ Skim-RNN achieves comparable result to Regular RNN with 2.3x FLOP-R on SQuAD.

Positive: I liked this movie, not because Tom Selleck was in it, but because it was a **good** story about baseball and it also had a semi-overheard *dramatized* view of some of the issues that a BASEBALL player coming to the end of their time in Major League sports must face. I also *greatly* enjoyed the cultural differences in American and Japanese baseball and the small facts on how the games are played differently.

Negative: Overall, it is a **good** movie to watch on Cable TV or rent on a cold winter’s night and watch about the ”Dog Days” of summer and know that spring training is only a few months away. A **good** movie for a baseball fan as well as a good “DATE” movie. Trust me on that one! **Wink**!

▲ Examples on IMDb. Skim-RNN skims black words and fully reads blue words.

Positive: I **loved** this movie, not because Tom Selleck was in it, but because it was a **good** story about baseball and it also had a semi-overheard dramatic view of some of the issues that a BASEBALL player coming to the end of their time in Major League sports must face. I also greatly enjoyed the cultural differences in American and Japanese baseball and the small facts on how the games are played differently.

Negative: Overall, it is a **good** movie to watch on Cable TV or rent on a cold winter’s night and watch about the ”Dog Days” of summer and know that spring training is only a few months away. A **good** movie for a baseball fan as well as a good “DATE” movie. Trust me on that one! **Wink**!

▲ Examples on SQuAD with Skim-RNN in different levels. Upper LSTMs skim more than lower LSTMs do.

▲ Trade-off between F1 and FLOP-R, obtained by adjusting the threshold for the skim. $d'=10$ & $d'=50$. 

---

**Table:**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Regular</th>
<th>LSTM-Jump</th>
<th>VCRNN</th>
<th>Skim-RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST</td>
<td>86.4</td>
<td>79.3</td>
<td>81.9</td>
<td>86.4</td>
</tr>
<tr>
<td>Rotten Tomatoes</td>
<td>82.5</td>
<td>79.3 / 1.6x Sp</td>
<td>81.4 / 1.6x Sp</td>
<td>84.2 / 1.3x Sp</td>
</tr>
<tr>
<td>IMdb</td>
<td>91.1</td>
<td>89.4 / 1.6x Sp</td>
<td>89.3 / 1.1x Sp</td>
<td>91.2 / 2.3x Sp</td>
</tr>
<tr>
<td>AGNews</td>
<td>93.5</td>
<td>89.3 / 1.1x Sp</td>
<td>89.3 / 1.1x Sp</td>
<td>93.6 / 1.0x Sp</td>
</tr>
</tbody>
</table>

FLOP-R (Float operation Reduction)