Simple Recurrent Units for Highly Parallelizable Recurrence

Lei et al (2018)

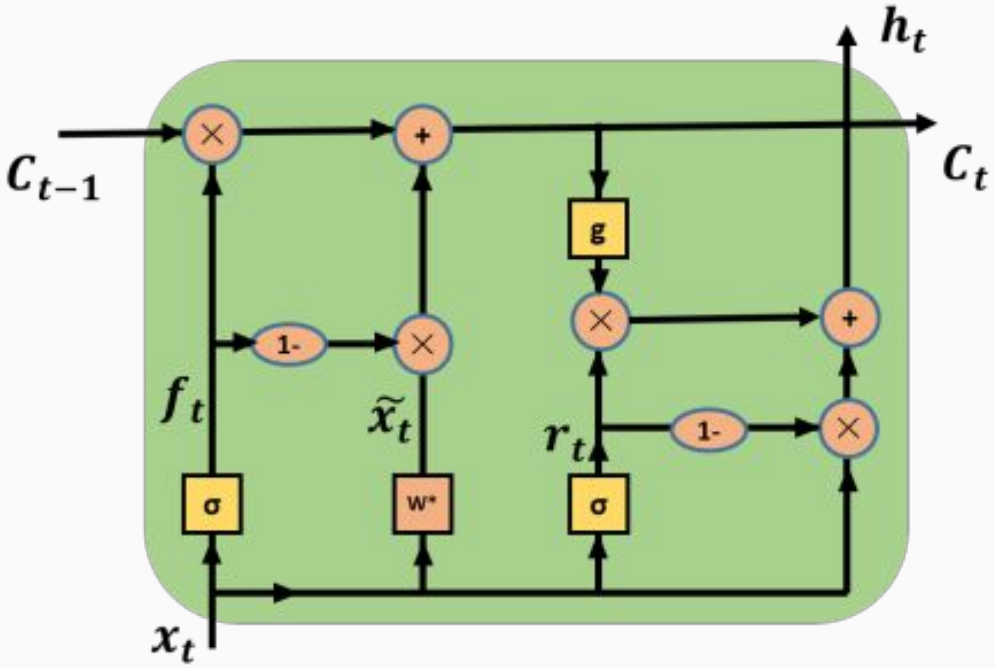
NLP reading club 2021/07/01

Key Points

- Make recurrent units parallelizable by replacing recurrent steps with pointwise multiplications instead of matrix multiplications
- Show training speeds comparable to CNNs
- Tested on text classification, Q&A, translation tasks-- in majority of cases, it shows improvement both on training time as compared to RNNs/LSTMs (interestingly, the paper does not report any comparison against Transformers, just that combined with a transformer architecture, it shows better translation perf)
-

SRU architecture

Simplified structure



$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{v}_f \odot \mathbf{c}_{t-1} + \mathbf{b}_f\right) \quad (1)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot (\mathbf{W}\mathbf{x}_t) \quad (2)$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r\mathbf{x}_t + \mathbf{v}_r \odot \mathbf{c}_{t-1} + \mathbf{b}_r\right) \quad (3)$$

$$\mathbf{h}_t = \mathbf{r}_t \odot \mathbf{c}_t + (1 - \mathbf{r}_t) \odot \mathbf{x}_t \quad (4)$$

https://bamtercelboo.github.io/2017/10/19/SRU-Summary/

Parallel training

Algorithm 1 Mini-batch version of the forward pass defined in Equations (1)–(4).

Indices: Sequence length L, mini-batch size B, hidden state dimension d.

Input: Input sequences batch $\mathbf{x}[l, i, j]$; grouped matrix multiplication $\mathbf{U}[l, i, j']$; initial state $\mathbf{c}_0[i, j]$; parameters $\mathbf{v}_f[j]$, $\mathbf{v}_r[j]$, $\mathbf{b}_f[j]$ and $\mathbf{b}_r[j]$.

Output: Output $\mathbf{h}[\cdot, \cdot, \cdot]$ and internal $\mathbf{c}[\cdot, \cdot, \cdot]$ states.

Initialize $\mathbf{h}[\cdot, \cdot, \cdot]$ and $\mathbf{c}[\cdot, \cdot, \cdot]$ as two $L \times B \times d$ tensors.

for $i = 1, \cdots, B; j = 1, \cdots, d$ **do** // **Parallelize each example** i **and dimension** j

$c = \mathbf{c}_0[i, j]$

for $l = 1, \cdots, L$ **do**

$f = \sigma\left(\mathbf{U}[l, i, j + d] + \mathbf{v}_f[j] \times c + \mathbf{b}_f[j]\right)$

$c = f \times c + (1 - f) \times \mathbf{U}[l, i, j]$

$r = \sigma\left(\mathbf{U}[l, i, j + d \times 2] + \mathbf{v}_r[j] \times c + \mathbf{b}_r[j]\right)$

$h = r \times c + (1 - r) \times \mathbf{x}[l, i, j]$

$\mathbf{c}[l, i, j] = c$

$\mathbf{h}[l, i, j] = h$

return $\mathbf{h}[\cdot, \cdot, \cdot]$ and $\mathbf{c}[\cdot, \cdot, \cdot]$

https://www.researchgate.net/figure/Basic-structure-of-the-Simple-Recurrent-Unit-SRU-Working-flow_fig1329740853

Experiment results -- processing times

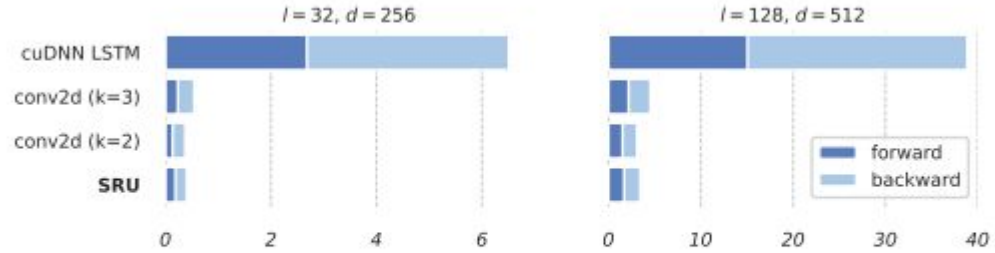


Figure 1: Average processing time in milliseconds of a batch of 32 samples using cuDNN LSTM, word-level convolution `conv2d` (with filter width $k = 2$ and $k = 3$), and the proposed SRU. We vary the number of tokens per sequence (l) and feature dimension (d).

Experimental results -- NLP tasks

Model	Size	CR	SUBJ	MR	TREC	MPQA	SST	Time
Best reported results:								
Wang and Manning (2013)		82.1	93.6	79.1	-	86.3	-	-
Kalchbrenner et al. (2014)		-	-	-	93.0	-	86.8	-
Kim (2014)		85.0	93.4	81.5	93.6	89.6	88.1	-
Zhang and Wallace (2017)		84.7	93.7	81.7	91.6	89.6	85.5	-
Zhao et al. (2015)		86.3	**95.5**	**83.1**	92.4	**93.3**	-	-
Our setup (default Adam, fixed word embeddings):								
CNN	360k	83.1±1.6	92.7±0.9	78.9±1.3	93.2±0.8	89.2±0.8	85.1±0.6	417
LSTM	352k	82.7±1.9	92.6±0.8	79.8±1.3	93.4±0.9	89.4±0.7	88.1±0.8	2409
QRNN (k=1)	165k	83.5±1.9	93.4±0.6	82.0±1.0	92.5±0.5	90.2±0.7	88.2±0.4	345
QRNN (k=1) + highway	204k	84.0±1.9	93.4±0.8	82.1±1.2	93.2±0.6	89.6±1.2	88.9±0.2	371
SRU (2 layers)	204k	84.9±1.6	93.5±0.6	82.3±1.2	94.0±0.5	90.1±0.7	89.2±0.3	**320**
SRU (4 layers)	303k	85.9±1.5	93.8±0.6	82.9±1.0	**94.8**±0.5	90.1±0.6	**89.6**±0.5	510
SRU (8 layers)	502k	**86.4**±1.7	93.7±0.6	**83.1**±1.0	94.7±0.5	90.2±0.8	88.9±0.6	879

Table 1: Test accuracies on classification benchmarks (Section 4.1). The first block presents best reported results of various methods. The second block compares SRU and other baselines given the same setup. For the SST dataset, we report average results of 5 runs. For other datasets, we perform 3 independent trials of 10-fold cross validation (3×10 runs). The last column compares the wall clock time (in seconds) to finish 100 epochs on the SST dataset.

MR: movie review sentiment
SUBJ: sentence subjectivity
CR: customer reviews polarity
TREC: question type
MPQA: opinion polarity
SST: Stanford sentiment treebank

Experimental results -- NLP tasks

Q&A

Model	# layers	Size	Dev EM	Dev F1	Time per epoch RNN	Time per epoch Total
LSTM (Chen et al., 2017)	3	4.1m	69.5	78.8	316s	431s
QRNN (k=1) + highway	4	2.4m	70.1 ± 0.1	79.4 ± 0.1	113s	214s
	6	3.2m	70.6 ± 0.1	79.6 ± 0.2	161s	262s
SRU	3	2.0m	**70.2** ± 0.3	**79.3** ± 0.1	58s	159s
SRU	4	2.4m	**70.7** ± 0.1	**79.7** ± 0.1	72s	173s
SRU	6	3.2m	**71.4** ± 0.1	**80.2** ± 0.1	100s	201s

Table 2: Exact match (EM) and F1 scores of various models on SQuAD (Section 4.2). We also report the total processing time per epoch and the time spent in RNN computations. SRU outperforms other models, and is more than five times faster than cuDNN LSTM.

Experimental results -- NLP tasks

translation : En-> German

Model	# layers	Size	BLEU score Valid	BLEU score Test	Speed (toks/sec)	Hours per epoch
Transformer (base)	6	76m	26.6±0.2 (26.9)	27.6±0.2 (27.9)	20k	2.0
Transformer (+SRU)	4	79m	26.7±0.1 (26.8)	27.8±0.1 (28.3)	**22k**	**1.8**
Transformer (+SRU)	5	90m	**27.1**±0.0 (**27.2**)	**28.3**±0.1 (**28.4**)	19k	2.1

Table 3: English→German translation results (Section 4.3). We perform 3 independent runs for each configuration. We select the best epoch based on the valid BLEU score for each run, and report the average results and the standard deviation over 3 runs. In addition, we experiment with averaging model checkpoints and use the averaged version for evaluation, following (Vaswani et al., 2017). We show the best BLEU results achieved in brackets.